aws

Panduan Developerr

OpenSearch Layanan Amazon

OpenSearch Layanan Amazon: Panduan Developerr

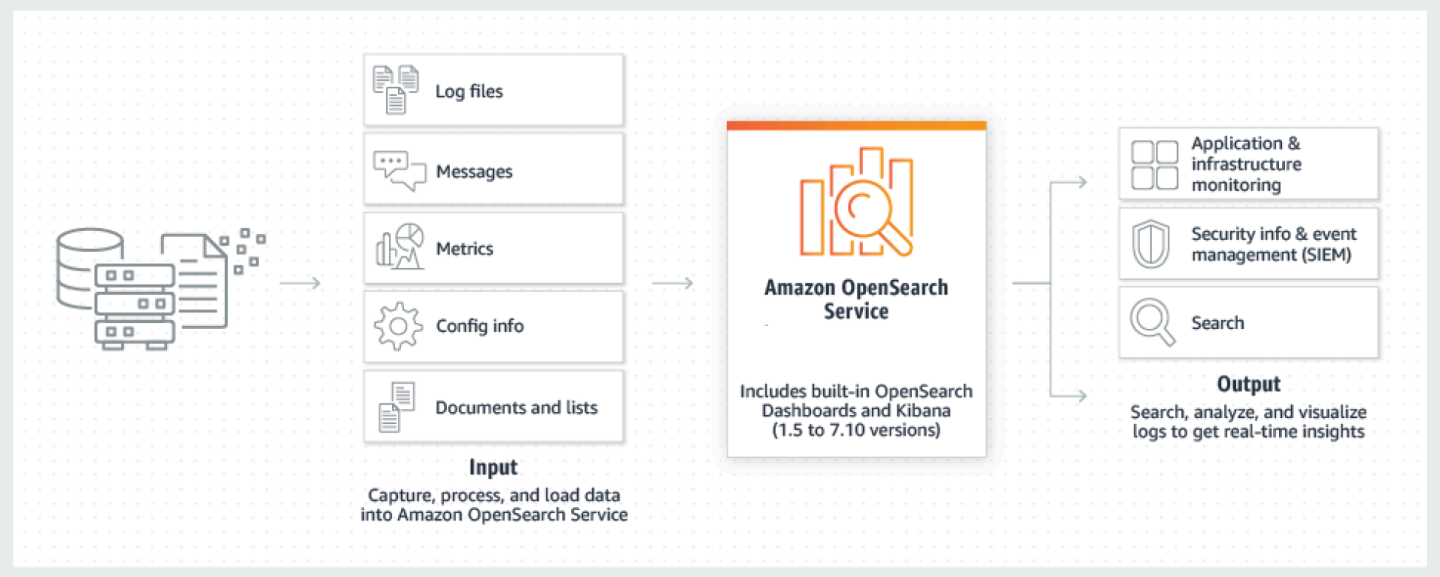
Table of Contents

Apa itu OpenSearch Layanan Amazon?

Amazon OpenSearch Service adalah layanan terkelola yang memudahkan penerapan, pengoperasian, dan skala OpenSearch cluster di AWS Cloud. Amazon OpenSearch Service mendukung OpenSearch dan warisan Elasticsearch OSS (hingga 7.10, versi open source terakhir dari perangkat lunak). Saat Anda membuat cluster, Anda memiliki opsi mesin pencari mana yang akan digunakan.

OpenSearchadalah mesin pencari dan analitik sumber terbuka sepenuhnya untuk kasus penggunaan seperti analitik log, pemantauan aplikasi waktu nyata, dan analisis clickstream. Untuk informasi lebih lanjut, lihat dokumentasi OpenSearch .

OpenSearch Layanan Amazon menyediakan semua sumber daya untuk OpenSearch klaster Anda dan meluncurkannya. Ini juga secara otomatis mendeteksi dan mengganti node OpenSearch Layanan yang gagal, mengurangi overhead yang terkait dengan infrastruktur yang dikelola sendiri. Anda dapat menskalakan klaster Anda dengan satu panggilan API atau beberapa klik di konsol.



Untuk mulai menggunakan OpenSearch Layanan, Anda membuat domain OpenSearch Layanan, yang setara dengan OpenSearch kluster. Setiap instans EC2 di cluster bertindak sebagai satu node OpenSearch Layanan.

Anda dapat menggunakan konsol OpenSearch Layanan untuk mengatur dan mengonfigurasi domain dalam hitungan menit. Jika Anda lebih suka akses terprogram, Anda dapat menggunakan AWS CLI atau SDK AWS.

Fitur OpenSearch Layanan Amazon

OpenSearch Layanan mencakup fitur-fitur berikut:

Skala

- Banyak konfigurasi CPU, memori, dan kapasitas penyimpanan yang dikenal sebagai tipe instans, termasuk instans Graviton yang hemat biaya
- Hingga 3 PB penyimpanan terlampir
- Hemat biaya UltraWarmdan penyimpanan dingin untuk data hanya-baca

Keamanan

- Kontrol akses AWS Identity and Access Management (IAM)
- Integrasi yang mudah dengan Amazon VPC dan grup keamanan VPC
- Enkripsi data saat istirahat dan node-to-node enkripsi
- Amazon Cognito, HTTP dasar, atau otentikasi SAMB untuk Dasbor OpenSearch
- Keamanan tingkat indeks, tingkat dokumen, dan tingkat bidang
- Log audit
- Dasbor multi-tenancy

Stabilitas

- Banyak lokasi geografis untuk sumber daya Anda, yang dikenal sebagai Wilayah dan Availability Zone
- Alokasi simpul di dua atau tiga Availability Zone di Wilayah AWS yang sama, yang dikenal sebagai Multi-AZ
- Simpul utama terdedikasi untuk membongkar tugas manajemen klaster
- Snapshot otomatis untuk mencadangkan dan memulihkan domain OpenSearch Layanan

Fleksibilitas

- Dukungan SQL untuk integrasi dengan aplikasi kecerdasan bisnis (BI)
- Paket kustom untuk meningkatkan hasil pencarian

Integrasi dengan layanan populer

- Visualisasi data menggunakan Dasbor OpenSearch
- Integrasi dengan Amazon CloudWatch untuk memantau metrik domain OpenSearch Layanan dan pengaturan alarm
- Integrasi dengan AWS CloudTrail untuk mengaudit panggilan API konfigurasi ke domain OpenSearch Layanan
- Integrasi dengan Amazon S3, Amazon Kinesis, dan Amazon DynamoDB untuk memuat data streaming ke Layanan OpenSearch
- Pemberitahuan dari Amazon SNS saat data Anda melebihi ambang batas tertentu

Amazon Tanpa OpenSearch Server

Amazon OpenSearch Serverless adalah konfigurasi on-demand, penskalaan otomatis, tanpa server untuk Amazon Service. OpenSearch Tanpa server menghilangkan kompleksitas operasional penyediaan, konfigurasi, dan penyetelan cluster Anda. OpenSearch Untuk informasi selengkapnya, lihat Amazon Tanpa OpenSearch Server.

OpenSearch Tertelan Amazon

Amazon OpenSearch Ingestion adalah pengumpul data yang dikelola sepenuhnya, didukung oleh Data Prepper, yang mengirimkan data log dan jejak waktu nyata ke domain Amazon OpenSearch Service dan koleksi Tanpa Server. OpenSearch Ini memungkinkan Anda untuk memfilter, memperkaya, mengubah, menormalkan, dan mengumpulkan data untuk analisis dan visualisasi hilir. Untuk informasi selengkapnya, lihat Amazon OpenSearch Ingestion.

Versi OpenSearch dan Elasticsearch yang didukung

OpenSearch Layanan saat ini mendukung OpenSearch versi berikut:

- 2.11, 2.9, 2.7, 2.5, 2.3, 1.3, 1.2, 1.1, 1.0

OpenSearch Layanan juga mendukung versi OSS Elasticsearch warisan berikut:

- 7.10, 7.9, 7.8, 7.7, 7.4, 7.1

- 6.8, 6.7, 6.5, 6.4, 6.3, 6.2, 6.0
- 5.6, 5.5, 5.3, 5.1
- 2.3
- 1.5

Untuk informasi lebih lanjut, lihat the section called “Operasi yang didukung”, the section called “Fitur berdasarkan versi mesin”, dan the section called “Plugin berdasarkan versi mesin”.

Jika Anda memulai proyek OpenSearch Layanan baru, kami sangat menyarankan Anda memilih OpenSearch versi terbaru yang didukung. Jika Anda memiliki domain yang ada yang menggunakan versi Elasticsearch lama, Anda dapat memilih untuk menyimpan domain atau memigrasikan data Anda. Untuk informasi selengkapnya, lihat the section called “Memutakhirkan domain”.

Harga untuk Amazon OpenSearch Service

Untuk OpenSearch Layanan, Anda membayar setiap jam penggunaan instans EC2 dan untuk ukuran kumulatif volume penyimpanan EBS yang melekat pada instans Anda. Biaya transfer data AWS standar juga berlaku.

Namun, ada beberapa pengecualian transfer data yang penting. Jika domain menggunakan beberapa Availability Zone, OpenSearch Layanan tidak mengenakan biaya untuk lalu lintas di antara Availability Zone. Transfer data yang signifikan terjadi dalam domain selama alokasi pecahan dan penyeimbangan kembali. OpenSearch Layanan baik meter maupun tagihan untuk lalu lintas ini. Demikian pula, OpenSearch Layanan tidak mengenakan biaya untuk transfer data UltraWarmantara/ node dingin dan Amazon S3.

Untuk detail harga selengkapnya, lihat harga Amazon OpenSearch Service. Untuk informasi tentang biaya yang timbul selama perubahan konfigurasi, lihat the section called “Biaya untuk perubahan konfigurasi”.

Memulai dengan Amazon OpenSearch Service

Untuk memulai, daftar untuk Akun AWS jika Anda belum memilikinya. Setelah Anda mengatur akun, selesaikan tutorial memulai untuk Amazon OpenSearch Service. Konsultasikan topik pengantar berikut jika Anda memerlukan informasi lebih lanjut sambil mempelajari tentang layanan ini:

- Buat domain

- Ukur domain dengan tepat untuk beban kerja Anda
- Kontrol akses ke domain Anda menggunakan kebijakan akses domain atau kontrol akses detail
- Mengindeks data secara manual atau dari layanan AWS lainnya
- Gunakan OpenSearch Dasbor untuk mencari data Anda dan membuat visualisasi

Untuk informasi tentang migrasi ke OpenSearch Layanan dari OpenSearch klaster yang dikelola sendiri, lihat. the section called “Migrasi keOpenSearchLayanan”

Layanan terkait

OpenSearch Layanan umumnya digunakan dengan layanan berikut:

Amazon CloudWatch

OpenSearch Domain layanan mengirim metrik secara otomatis CloudWatch sehingga Anda dapat memantau kesehatan dan kinerja domain. Untuk informasi selengkapnya, lihat Memantau metrik OpenSearch klaster dengan Amazon CloudWatch.

CloudWatch Log juga bisa pergi ke arah lain. Anda dapat mengonfigurasi CloudWatch Log untuk mengalirkan data ke OpenSearch Layanan untuk analisis. Untuk mempelajari selengkapnya, lihat the section called “Memuat data streaming dari Amazon CloudWatch”.

AWS CloudTrail

Gunakan AWS CloudTrail untuk mendapatkan riwayat panggilan API konfigurasi OpenSearch Layanan dan peristiwa terkait untuk akun Anda. Untuk informasi selengkapnya, lihat Pemantauan panggilan API Amazon OpenSearch Service dengan AWS CloudTrail.

Amazon Kinesis

Kinesis adalah layanan terkelola untuk pemrosesan data streaming secara waktu nyata dalam skala besar. Untuk informasi lebih lanjut, lihat the section called “Memuat data streaming dari Amazon Kinesis Data Streams” dan the section called “Memuat data streaming dari Amazon Data Firehose”.

Amazon S3

Amazon Simple Storage Service (Amazon S3) menyediakan penyimpanan untuk internet. Panduan ini menyediakan kode sampel Lambda untuk integrasi dengan Amazon S3. Untuk informasi selengkapnya, lihat the section called “Memuat data streaming dari Amazon S3”.

AWSIAM

AWS Identity and Access Management(IAM) adalah layanan web yang dapat Anda gunakan untuk mengelola akses ke domain OpenSearch Layanan Anda. Untuk informasi selengkapnya, lihat the section called “Manajemen Identitas dan Akses”.

AWS Lambda

AWS Lambda adalah layanan komputasi yang memungkinkan Anda menjalankan kode tanpa perlu menyediakan atau mengelola server. Panduan ini menyediakan kode sampel Lambda untuk pengaliran data dari DynamoDB, Amazon S3, dan Kinesis. Untuk informasi selengkapnya, lihat the section called “Memuat data streaming ke OpenSearch Layanan”.

Amazon DynamoDB

Amazon DynamoDB adalah layanan basis data NoSQL terkelola penuh yang menyediakan performa cepat dan dapat diprediksi dengan skalabilitas mulus. Untuk mempelajari lebih lanjut tentang streaming data ke OpenSearch Layanan, lihatthe section called “Memuat data streaming dari Amazon DynamoDB”.

Amazon QuickSight

Anda dapat memvisualisasikan data dari OpenSearch Layanan menggunakan QuickSight dasbor Amazon. Untuk informasi selengkapnya, lihat Menggunakan OpenSearch Layanan Amazon dengan Amazon QuickSight di Panduan QuickSight Pengguna Amazon.

> **Note**
>
> OpenSearch termasuk kode Elasticsearch berlisensi Apache tertentu dari Elasticsearch BV dan kode sumber lainnya. Elasticsearch BV bukanlah sumber dari kode sumber lainnya. ELASTICSEARCH adalah merek dagang terdaftar dari Elasticsearch B.V.

Menyiapkan OpenSearch Layanan Amazon

Topik

Daftar Akun AWS

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

Untuk mendaftar 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 Akun AWS, 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 akan mengirimkan email konfirmasi kepada Anda 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 mendaftarAkun AWS, amankanPengguna root akun AWS, aktifkanAWS IAM Identity Center, dan buat pengguna administratif sehingga Anda tidak menggunakan pengguna root untuk tugas sehari-hari.

Mengamankan Pengguna root akun AWS Anda

1. Masuk ke AWS Management Console sebagai pemilik akun dengan memilih Pengguna root dan memasukkan alamat email Akun AWS Anda. Pada halaman berikutnya, masukkan kata sandi Anda.

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

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

Membuat pengguna administratif

1. Aktifkan Pusat Identitas IAM.

   Untuk petunjuk, lihat Mengaktifkan AWS IAM Identity Center di Panduan AWS IAM Identity Center Pengguna.
2. Di Pusat Identitas IAM, berikan akses administratif ke pengguna administratif.

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

Masuk sebagai pengguna 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 akses AWS dalam Panduan Pengguna AWS Sign-In.

Berikan izin

Di lingkungan produksi, kami menyarankan Anda menggunakan kebijakan yang lebih halus. Untuk mempelajari lebih lanjut tentang manajemen akses, lihat Manajemen akses untuk AWS sumber daya di Panduan Pengguna IAM.

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

- Pengguna dan grup di AWS IAM Identity Center:

  Buat set izin. Ikuti instruksi di Buat set izin di Panduan Pengguna AWS IAM Identity Center.

- Pengguna yang dikelola di IAM melalui penyedia identitas:

  Buat peran untuk federasi identitas. Ikuti instruksi dalam Membuat peran untuk penyedia identitas pihak ketiga (federasi) di Panduan Pengguna IAM.

- Pengguna IAM:
  - Buat peran yang dapat diambil pengguna Anda. Ikuti instruksi dalam Membuat peran untuk pengguna IAM dalam Panduan Pengguna IAM.
  - (Tidak disarankan) Lampirkan kebijakan langsung ke pengguna atau tambahkan pengguna ke grup pengguna. Ikuti instruksi dalam Menambahkan izin ke pengguna (konsol) dalam Panduan Pengguna IAM.

Berikan akses terprogram

Pengguna membutuhkan akses terprogram jika mereka ingin berinteraksi dengan AWS di luar AWS Management Console. Cara memberikan akses terprogram bergantung pada jenis pengguna yang mengakses AWS.

Untuk memberi pengguna akses terprogram, pilih salah satu opsi berikut.

Pengguna mana yang membutuhkan akses terprogram?	Ke	Oleh
Identitas tenaga kerja (Pengguna yang dikelola di Pusat Identitas IAM)	Gunakan kredensial sementara untuk menandatangani permintaan terprogram ke AWS CLI, SDK AWS, atau API AWS.	Mengikuti petunjuk untuk antarmuka yang ingin Anda gunakan. • Untuk AWS CLI, lihat Mengonfigurasi AWS CLI untuk menggunakan AWS IAM Identity Center di

Pengguna mana yang membutuhkan akses terprogram?	Ke	Oleh
		Panduan Pengguna AWS Command Line Interface. • Untuk SDK AWS, alat, dan API AWS, lihat Autentikasi Pusat Identitas IAM di Panduan Referensi SDK dan Alat AWS.
IAM	Gunakan kredensial sementara untuk menandatangani permintaan terprogram ke AWS CLI, SDK AWS, atau API AWS.	Ikuti petunjuk dalam Menggunakan kredensial sementara dengan sumber daya AWS di Panduan Pengguna IAM.

Pengguna mana yang membutuhkan akses terprogram?	Ke	Oleh
IAM	(Tidak disarankan) Gunakan kredensial jangka panjang untuk menandatangani permintaan terprogram ke AWS CLI, AWS SDK, atau API AWS.	Mengikuti petunjuk untuk antarmuka yang ingin Anda gunakan. • Untuk AWS CLI, lihat Mengautentikasi menggunakan kredensial pengguna IAM di Panduan Pengguna AWS Command Line Interface. • Untuk SDK dan alat AWS, lihat Mengautentikasi menggunakan kredensial jangka panjang di Panduan Referensi SDK dan Alat AWS. • Untuk API AWS, lihat Mengelola kunci akses untuk pengguna IAM di Panduan Pengguna IAM.

Instal dan konfigurasikan AWS CLI

Jika Anda ingin menggunakan OpenSearch Service API, Anda harus menginstal versi terbaru dari AWS Command Line Interface (AWS CLI). Anda tidak AWS CLI perlu menggunakan OpenSearch Layanan dari konsol, dan Anda dapat memulai tanpa CLI dengan mengikuti langkah-langkah di. Memulai dengan AmazonOpenSearchLayanan

Untuk mengatur AWS CLI

1. Untuk menginstal versi terbaru dari macOS, Linux, atau Windows, lihat Menginstal atau memperbarui versi terbaru dari file. AWS CLIAWS CLI

2. Untuk mengonfigurasi AWS CLI dan mengamankan pengaturan akses AndaLayanan AWS, termasuk OpenSearch Layanan, lihat Konfigurasi cepat dengan `aws configure`.
3. Untuk memverifikasi pengaturan, masukkan DataBrew perintah berikut pada prompt perintah.

```
aws opensearch help
```

AWS CLIperintah menggunakan default Wilayah AWS dari konfigurasi Anda, kecuali jika Anda mengaturnya dengan parameter atau profil. Untuk mengatur Anda Wilayah AWS dengan parameter, Anda dapat menambahkan `--region` parameter ke setiap perintah.

Untuk mengatur profil AndaWilayah AWS, pertama-tama tambahkan profil bernama dalam `~/.aws/config` file atau `%UserProfile%/.aws/config` file (untuk Microsoft Windows). Ikuti langkah-langkah di Profil bernama untuk AWS CLI. Selanjutnya, atur pengaturan Anda Wilayah AWS dan lainnya dengan perintah yang mirip dengan yang ada di contoh berikut.

```
[profile opensearch]
aws_access_key_id = ACCESS-KEY-ID-OF-IAM-USER
aws_secret_access_key = SECRET-ACCESS-KEY-ID-OF-IAM-USER
region = us-east-1
output = text
```

Buka konsol

Sebagian besar topik berorientasi konsol di bagian ini dimulai dari konsol Layanan. OpenSearch Jika Anda belum masukAkun AWS, masuk, lalu buka konsol OpenSearch Layanan dan lanjutkan ke bagian berikutnya untuk melanjutkan memulai OpenSearch Layanan.

Memulai dengan AmazonOpenSearchLayanan

Tutorial ini menunjukkan cara menggunakan AmazonOpenSearchLayanan untuk membuat dan mengkonfigurasi domain uji. SebuahOpenSearchDomain layanan identik denganOpenSearchcluster. Domain adalah klaster dengan pengaturan, tipe instans, jumlah instans, dan sumber daya penyimpanan yang Anda tentukan.

Tutorial ini memandu Anda melalui langkah-langkah dasar untuk mendapatkanOpenSearchDomain layanan aktif dan berjalan dengan cepat. Untuk informasi selengkapnya, lihat Membuat dan mengelola domain dan topik lainnya dalam panduan ini. Untuk informasi tentang migrasi keOpenSearchLayanan dari yang dikelola sendiriOpenSearchcluster, lihatthe section called “Migrasi keOpenSearchLayanan”.

Anda dapat menyelesaikan langkah-langkah dalam tutorial ini dengan menggunakanOpenSearchKonsol layanan,AWS CLI, atauAWSSDK. Untuk informasi tentang menginstal dan menyiapkan AWS CLI, lihat AWS Command Line InterfacePanduan Pengguna.

Langkah 1: Buat AmazonOpenSearchDomain layanan

> **Important**
>
> Ini adalah tutorial ringkas untuk mengkonfigurasiujiAmazonOpenSearchDomain layanan. Jangan gunakan proses ini untuk membuat domain produksi. Untuk versi komprehensif dari proses yang sama, lihat Membuat dan mengelola domain.

SebuahOpenSearchDomain layanan identik denganOpenSearchcluster. Domain adalah klaster dengan pengaturan, tipe instans, jumlah instans, dan sumber daya penyimpanan yang Anda tentukan. Anda dapat membuatOpenSearchLayanan domain dengan menggunakan konsol,AWS CLI, atauAWSSDK.

Untuk membuatOpenSearchDomain layanan menggunakan konsol

1. Masuk ke https://aws.amazon.com, dan pilih Masuk ke Konsol.
2. Di bawahAnalitik, pilihAmazonOpenSearchLayanan.
3. Pilih Create domain (Buat domain).

4. Memberikan nama untuk domain. Contoh-contoh dalam tutorial ini menggunakan nama film.
5. Untuk metode pembuatan domain, pilihBuat standar.

> **Note**
>
> Untuk mengonfigurasi domain produksi dengan cepat dengan praktik terbaik, Anda dapat memilihMudah membuat. Untuk tujuan pengembangan dan pengujian tutorial ini, kita akan menggunakanBuat standar.

6. Untuk template, pilihDev/uji.
7. Untuk opsi penyebaran, pilihDomain dengan siaga.
8. UntukVersi, pilih versi terbaru.
9. Untuk saat ini, abaikanNode data,Penyimpanan data yang hangat dan dingin,Node master khusus,Konfigurasi snapshot, danEndpoint kustombagian.
10. Untuk kesederhanaan dalam tutorial ini, gunakan domain akses publik. Di bawahJaringan, pilihAkses publik.
11. Dalam pengaturan kontrol akses berbutir halus, simpanAktifkan kontrol akses berbutir haluskotak centang yang dipilih. PilihBuat pengguna masterdan berikan nama pengguna dan kata sandi.
12. Untuk saat ini, abaikan Autentikasi SAML dan bagian Autentikasi Amazon Cognito.
13. UntukKebijakan akses, pilihHanya gunakan kontrol akses berbutir halus. Dalam tutorial ini, kontrol akses detail menangani autentikasi, bukan kebijakan akses domain.
14. Abaikan pengaturan lainnya dan pilihBuat. Domain baru biasanya membutuhkan waktu 15-30 menit untuk menginisialisasi, tetapi dapat memakan waktu lebih lama tergantung pada konfigurasi. Setelah domain Anda menginisialisasi, pilih untuk membuka panel konfigurasinya. Perhatikan endpoint domain di bawahInformasi Umum(misalnya,`https://search-`*`my-domain`*`.us-east-1.es.amazonaws.com`), yang akan Anda gunakan pada langkah berikutnya.

Berikutnya:Unggah data keOpenSearchDomain layanan untuk pengindeksan

Langkah 2: Unggah data ke AmazonOpenSearchLayanan untuk pengindeksan

> **Important**
>
> Ini adalah tutorial ringkas untuk mengunggah sejumlah kecil data pengujian ke AmazonOpenSearchLayanan. Untuk lebih lanjut tentang mengunggah data dalam domain produksi, lihat Mengindeks data.

Anda dapat mengunggah data keOpenSearchDomain layanan menggunakan baris perintah atau sebagian besar bahasa pemrograman.

Permintaan contoh berikut menggunakan curl (klien HTTP umum) untuk singkat dan kenyamanan. Klien seperti curl tidak dapat melakukan penandatanganan permintaan yang diperlukan jika kebijakan akses Anda menentukan pengguna atau IAM role. Agar berhasil menyelesaikan proses ini, Anda harus menggunakan kontrol akses berbutir halus dengan nama pengguna dan kata sandi utama seperti yang Anda konfigurasikanLangkah 1.

Anda dapat menginstal curl di Windows dan menggunakannya dari command prompt, tetapi kami merekomendasikan alat seperti Cygwin atau Subsistem Windows untuk Linux. macOS dan sebagian besar distribusi Linux dilengkapi dengan curl yang sudah diinstal sebelumnya.

Opsi 1: Unggah satu dokumen

Jalankan perintah berikut untuk menambahkan dokumen tunggal ke domain film:

```
curl -XPUT -u 'master-user:master-user-password' 'domain-endpoint/movies/_doc/1' -d
 '{"director": "Burton, Tim", "genre": ["Comedy","Sci-Fi"], "year": 1996, "actor":
 ["Jack Nicholson","Pierce Brosnan","Sarah Jessica Parker"], "title": "Mars Attacks!"}'
 -H 'Content-Type: application/json'
```

Dalam perintah, berikan nama pengguna dan kata sandi yang Anda buatLangkah 1.

Untuk penjelasan rinci tentang perintah ini dan cara membuat permintaan yang ditandatanganiOpenSearchLayanan, lihatMengindeks data.

Opsi 2: Unggah beberapa dokumen

Untuk mengunggah file JSON yang berisi beberapa dokumen keOpenSearchDomain layanan

1. Buat file lokal bernamabulk_movies.json. Tempel konten berikut ke dalam file dan tambahkan baris baru tambahan:

```
{ "index" : { "_index": "movies", "_id" : "2" } }
{"director": "Frankenheimer, John", "genre": ["Drama", "Mystery", "Thriller",
 "Crime"], "year": 1962, "actor": ["Lansbury, Angela", "Sinatra, Frank", "Leigh,
 Janet", "Harvey, Laurence", "Silva, Henry", "Frees, Paul", "Gregory, James",
 "Bissell, Whit", "McGiver, John", "Parrish, Leslie", "Edwards, James", "Flowers,
 Bess", "Dhiegh, Khigh", "Payne, Julie", "Kleeb, Helen", "Gray, Joe", "Nalder,
 Reggie", "Stevens, Bert", "Masters, Michael", "Lowell, Tom"], "title": "The
 Manchurian Candidate"}
{ "index" : { "_index": "movies", "_id" : "3" } }
{"director": "Baird, Stuart", "genre": ["Action", "Crime", "Thriller"], "year":
 1998, "actor": ["Downey Jr., Robert", "Jones, Tommy Lee", "Snipes, Wesley",
 "Pantoliano, Joe", "Jacob, Ir\u00e8ne", "Nelligan, Kate", "Roebuck, Daniel",
 "Malahide, Patrick", "Richardson, LaTanya", "Wood, Tom", "Kosik, Thomas",
 "Stellate, Nick", "Minkoff, Robert", "Brown, Spitfire", "Foster, Reese",
 "Spielbauer, Bruce", "Mukherji, Kevin", "Cray, Ed", "Fordham, David", "Jett,
 Charlie"], "title": "U.S. Marshals"}
{ "index" : { "_index": "movies", "_id" : "4" } }
{"director": "Ray, Nicholas", "genre": ["Drama", "Romance"], "year": 1955, "actor":
 ["Hopper, Dennis", "Wood, Natalie", "Dean, James", "Mineo, Sal", "Backus, Jim",
 "Platt, Edward", "Ray, Nicholas", "Hopper, William", "Allen, Corey", "Birch,
 Paul", "Hudson, Rochelle", "Doran, Ann", "Hicks, Chuck", "Leigh, Nelson",
 "Williams, Robert", "Wessel, Dick", "Bryar, Paul", "Sessions, Almira", "McMahon,
 David", "Peters Jr., House"], "title": "Rebel Without a Cause"}
```

2. Jalankan perintah berikut di direktori lokal tempat file disimpan untuk mengunggahnya kefilmdomain:

```
curl -XPOST -u 'master-user:master-user-password' 'domain-endpoint/_bulk' --data-
binary @bulk_movies.json -H 'Content-Type: application/json'
```

Untuk informasi lebih lanjut tentang format file dalam jumlah besar, lihat Mengindeks data.

Selanjutnya:Cari dokumen

Langkah 3: Cari dokumen di AmazonOpenSearchLayanan

Untuk mencari dokumen di AmazonOpenSearchDomain layanan, gunakanOpenSearchAPI pencarian. Atau, Anda dapat menggunakanOpenSearchDasboruntuk mencari dokumen di domain.

Cari dokumen dari baris perintah

Jalankan perintah berikut untuk mencari domain film untuk kata mars:

```
curl -XGET -u 'master-user:master-user-password' 'domain-endpoint/movies/_search?
q=mars&pretty=true'
```

Jika Anda menggunakan data dalam jumlah besar di halaman sebelumnya, coba cari rebel.

Anda akan melihat respons yang mirip dengan berikut ini:

```
{
  "took" : 5,
  "timed_out" : false,
  "_shards" : {
    "total" : 5,
    "successful" : 5,
    "skipped" : 0,
    "failed" : 0
  },
  "hits" : {
    "total" : {
      "value" : 1,
      "relation" : "eq"
    },
    "max_score" : 0.2876821,
    "hits" : [
      {
        "_index" : "movies",
        "_type" : "_doc",
        "_id" : "1",
        "_score" : 0.2876821,
        "_source" : {
          "director" : "Burton, Tim",
          "genre" : [
            "Comedy",
            "Sci-Fi"
          ],
```

```
          "year" : 1996,
          "actor" : [
            "Jack Nicholson",
            "Pierce Brosnan",
            "Sarah Jessica Parker"
          ],
          "title" : "Mars Attacks!"
        }
      }
    ]
  }
}
```

Cari dokumen menggunakanOpenSearchDasbor

OpenSearchDasbor adalah alat visualisasi open source populer yang dirancang untuk digunakanOpenSearch. Ini menyediakan antarmuka pengguna yang berguna bagi Anda untuk mencari dan memantau indeks Anda.

Untuk mencari dokumen dariOpenSearchDomain layanan menggunakan Dasbor

1. Arahkan keOpenSearchURL dasbor untuk domain Anda. Anda dapat menemukan URL di dasbor domain diOpenSearchKonsol layanan. URL mengikuti format ini:

   ```
   domain-endpoint/_dashboards/
   ```

2. Masuk menggunakan nama pengguna dan kata sandi utama Anda.
3. Untuk menggunakan Dasbor, Anda perlu membuat setidaknya satu pola indeks. Dasbor menggunakan pola-pola ini untuk mengidentifikasi indeks mana yang ingin Anda analisis. Buka panel navigasi kiri, pilihManajemen Stack, pilihPola Indeks, dan kemudian pilihBuat pola indeks. Untuk tutorial ini, masukkan film.
4. Pilih Langkah selanjutnya dan kemudian pilih Buat pola indeks. Setelah pola dibuat, Anda dapat melihat berbagai bidang dokumen seperti `actor` dan `director`.
5. Kembali kePola Indekshalaman dan pastikan bahwa`movies`diatur sebagai default. Jika tidak, pilih pola dan pilih ikon bintang untuk menjadikannya default.
6. Untuk mulai mencari data Anda, buka panel navigasi kiri lagi dan pilihTemukan.
7. Di bilah pencarian, masukkan mars jika Anda mengunggah satu dokumen, atau rebel jika Anda mengunggah beberapa dokumen, lalu tekan Masukkan. Anda dapat mencoba mencari istilah lain, seperti nama aktor atau sutradara.

Selanjutnya:Hapus domain

Langkah 4: Menghapus AmazonOpenSearchDomain layanan

Karena domain film dari tutorial ini adalah untuk tujuan pengujian, pastikan untuk menghapusnya ketika Anda selesai bereksperimen untuk menghindari menimbulkan biaya.

Untuk menghapusOpenSearchDomain layanan dari konsol

1. Masuk keAmazonOpenSearchLayanankonsol.
2. Di bawahDomain, pilihfilmdomain.
3. PilihHapusdan konfirmasi penghapusan.

Langkah selanjutnya

Sekarang setelah Anda tahu cara membuat domain dan indeks data, Anda mungkin ingin mencoba beberapa latihan berikut:

- Mempelajari tentang opsi lanjutan lainnya untuk membuat domain. Untuk informasi selengkapnya, lihat Membuat dan mengelola domain.
- Temukan cara mengelola indeks di domain Anda. Untuk informasi selengkapnya, lihat Mengelola indeks.
- Cobalah salah satu tutorial untuk bekerja dengan AmazonOpenSearchLayanan. Untuk informasi selengkapnya, lihat Tutorial.

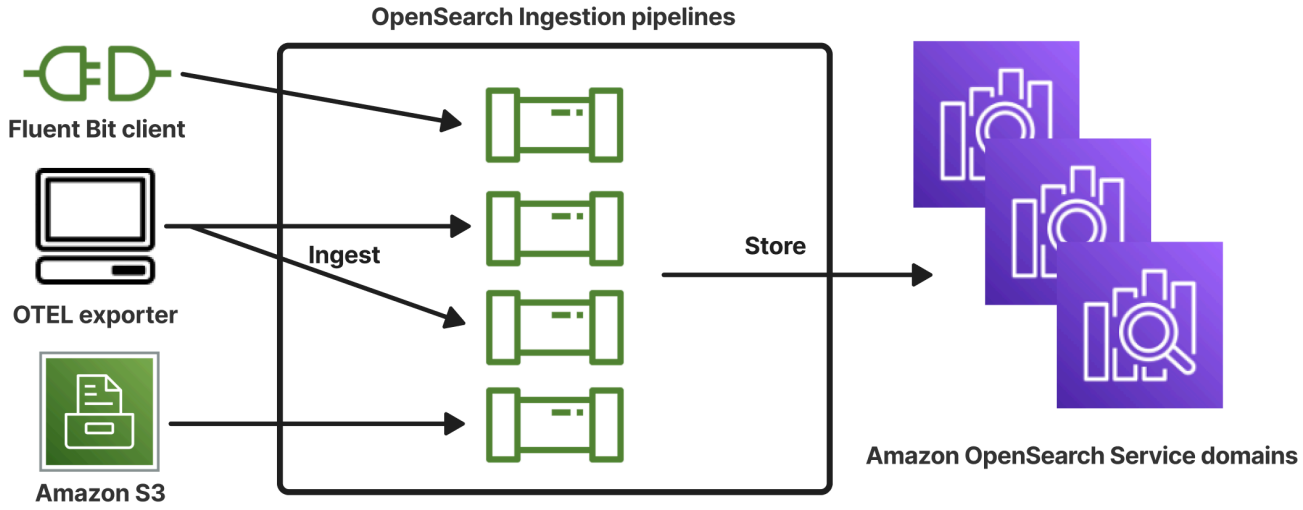
OpenSearch Tertelan Amazon

Amazon OpenSearch Ingestion adalah pengumpul data tanpa server yang dikelola sepenuhnya yang mengirimkan data log, metrik, dan jejak waktu nyata ke domain OpenSearch Layanan Amazon dan koleksi Tanpa Server. OpenSearch

Dengan OpenSearch Ingestion, Anda tidak perlu lagi menggunakan solusi pihak ketiga seperti Logstash atau Jaeger untuk menyerap data ke dalam domain Layanan dan koleksi Tanpa Server Anda. OpenSearch OpenSearch Anda mengonfigurasi produsen data Anda untuk mengirim data ke OpenSearch Ingestion. Kemudian, secara otomatis mengirimkan data ke domain atau koleksi yang Anda tentukan. Anda juga dapat mengonfigurasi OpenSearch Ingestion untuk mengubah data Anda sebelum mengirimkannya.

Selain itu, dengan OpenSearch Ingestion, Anda tidak perlu khawatir tentang penyediaan server, mengelola dan menambal perangkat lunak, atau menskalakan cluster server Anda. Anda menyediakan saluran pipa konsumsi langsung di dalam AWS Management Console, dan OpenSearch Ingestion menangani pengelolaan dan penskalaannya.

OpenSearch Ingestion adalah bagian dari Amazon Service. OpenSearch Ini didukung oleh Data Prepper, yang merupakan pengumpul data open source yang dapat memfilter, memperkaya, mengubah, menormalkan, dan mengumpulkan data untuk analisis dan visualisasi hilir.



Topik

- Konsep utama
- Manfaat OpenSearch Tertelan
- Batasan

Konsep utama

Saat Anda memulai dengan OpenSearch Ingestion, Anda bisa mendapatkan keuntungan dari memahami konsep-konsep berikut:

Alur

Dari perspektif OpenSearch Ingestion, pipeline mengacu pada pengumpul data tunggal yang disediakan yang Anda buat dalam Layanan. OpenSearch Anda dapat menganggapnya sebagai

keseluruhan file konfigurasi YAMB, yang mencakup satu atau lebih sub-pipeline. Untuk langkah-langkah untuk membuat saluran konsumsi, lihat. the section called “Membuat jaringan pipa”

Sub-pipa

Anda menentukan sub-pipeline dalam file konfigurasi YAMB. Setiap sub-pipeline adalah kombinasi dari sumber, buffer, nol atau lebih prosesor, dan satu atau lebih sink. Anda dapat menentukan beberapa sub-pipeline dalam satu file YAMG, masing-masing dengan sumber, prosesor, dan sink unik. Untuk membantu dalam pemantauan dengan CloudWatch dan layanan lainnya, kami sarankan Anda menentukan nama pipeline yang berbeda dari semua sub-pipeline-nya.

Anda dapat merangkai beberapa sub-pipeline bersama-sama dalam satu file YAMB, sehingga sumber untuk satu sub-pipeline adalah sub-pipeline lain, dan sink adalah sub-pipeline ketiga. Sebagai contoh, lihat the section called “OpenTelemetry Kolektor”.

Sumber

Komponen input dari sub-pipeline. Ini mendefinisikan mekanisme di mana pipa mengkonsumsi catatan. Sumber dapat mengkonsumsi peristiwa baik dengan menerimanya melalui HTTPS, atau dengan membaca dari titik akhir eksternal seperti Amazon S3. Ada dua jenis sumber: berbasis push dan pull-based. Sumber berbasis push, seperti log HTTP dan OTel, mengalirkan catatan ke titik akhir konsumsi. Sumber berbasis tarik, seperti OtEL trace dan S3, menarik data dari sumbernya.

Prosesor

Unit pemrosesan menengah yang dapat memfilter, mengubah, dan memperkaya catatan ke dalam format yang diinginkan sebelum menerbitkannya ke wastafel. Prosesor adalah komponen opsional dari pipa. Jika Anda tidak mendefinisikan prosesor, catatan dipublikasikan dalam format yang ditentukan dalam sumber. Anda dapat memiliki lebih dari satu prosesor. Pipeline menjalankan prosesor dalam urutan yang Anda definisikan.

Wastafel

Komponen output dari sub-pipeline. Ini mendefinisikan satu atau lebih tujuan yang sub-pipeline menerbitkan catatan. OpenSearch Ingestion mendukung domain OpenSearch Layanan sebagai sink. Ini juga mendukung sub-pipeline sebagai sink. Ini berarti Anda dapat merangkai beberapa sub-pipeline dalam satu pipa OpenSearch Ingestion (file YAMB). OpenSearch Cluster yang dikelola sendiri tidak didukung sebagai sink.

Penyangga

Bagian dari prosesor yang bertindak sebagai lapisan antara sumber dan wastafel. Anda tidak dapat mengonfigurasi buffer secara manual di dalam pipeline Anda. OpenSearch Ingestion menggunakan konfigurasi buffer default.

Rute

Bagian dari prosesor yang memungkinkan pembuat pipeline hanya mengirim peristiwa yang sesuai dengan kondisi tertentu ke sink yang berbeda.

Definisi sub-pipeline yang valid harus berisi sumber dan wastafel. Untuk informasi selengkapnya tentang masing-masing elemen pipeline ini, lihat referensi konfigurasi.

Manfaat OpenSearch Tertelan

OpenSearch Tertelan memiliki manfaat utama sebagai berikut:

- Menghilangkan kebutuhan bagi Anda untuk mengelola pipa yang disediakan sendiri secara manual.
- Secara otomatis menskalakan saluran pipa Anda berdasarkan batas kapasitas yang Anda tentukan.
- Selalu perbarui pipeline Anda dengan patch keamanan dan bug.
- Menyediakan opsi untuk menghubungkan saluran pipa ke virtual private cloud (VPC) Anda untuk lapisan keamanan tambahan.
- Memungkinkan Anda menghentikan dan memulai jaringan pipa untuk mengontrol biaya.
- Menyediakan cetak biru konfigurasi pipeline untuk kasus penggunaan populer untuk membantu Anda bangun dan berjalan lebih cepat.
- Memungkinkan Anda berinteraksi secara terprogram dengan pipeline Anda melalui berbagai AWS SDK dan API Ingestion. OpenSearch
- Mendukung pemantauan kinerja di Amazon CloudWatch dan pencatatan kesalahan di CloudWatch Log.

Batasan

OpenSearch Tertelan memiliki keterbatasan sebagai berikut:

- Anda hanya dapat menyerap data ke dalam domain yang menjalankan OpenSearch 1.0 atau yang lebih baru, atau Elasticsearch 6.8 atau yang lebih baru. Jika Anda menggunakan sumber jejak OTel, sebaiknya gunakan Elasticsearch 7.9 atau yang lebih baru agar Anda dapat menggunakan plugin Dasbor. OpenSearch
- Jika pipeline menulis ke domain OpenSearch Layanan yang ada di dalam VPC, pipeline harus dibuat Wilayah AWS sama dengan domain.
- Anda hanya dapat mengonfigurasi satu sumber data dalam definisi pipeline.
- Anda tidak dapat menentukan OpenSearch cluster yang dikelola sendiri sebagai sink.
- Anda tidak dapat menentukan titik akhir kustom sebagai wastafel. Anda masih dapat menulis ke domain yang memiliki titik akhir kustom diaktifkan, tetapi Anda harus menentukan titik akhir standarnya.
- Anda tidak dapat menentukan sumber daya dalam Wilayah keikutsertaan sebagai sumber atau sink.
- Ada beberapa kendala pada parameter yang dapat Anda sertakan dalam konfigurasi pipeline. Untuk informasi selengkapnya, lihat the section called “Persyaratan dan kendala konfigurasi”.

Versi Data Prepper yang Didukung

OpenSearch Ingestion saat ini mendukung versi utama Data Prepper berikut:

- 2.x

Saat Anda membuat pipeline, gunakan `version` opsi yang diperlukan untuk menentukan versi utama Data Prepper yang akan digunakan. Sebagai contoh,`version: "2"`. OpenSearch Ingestion mengambil versi minor terbaru yang didukung dari versi utama itu dan menyediakan pipeline dengan versi itu. Untuk informasi selengkapnya, lihat the section called “Menentukan versi pipeline”.

Saat ini, saluran pipa OpenSearch Ingestion disediakan dengan versi 2.7 dari Data Prepper. Untuk informasi, lihat catatan rilis 2.7. Untuk informasi tentang fitur dan perbaikan bug yang ada di setiap versi Data Prepper, lihat halaman Rilis. Tidak setiap versi minor dari versi utama tertentu didukung oleh OpenSearch Ingestion.

Saat Anda memperbarui file konfigurasi YAMB pipeline, jika ada dukungan untuk versi minor baru dari Data Prepper, OpenSearch Ingestion akan secara otomatis memutakhirkan pipeline ke versi minor terbaru yang didukung dari versi utama yang ditentukan dalam konfigurasi pipeline. Misalnya, Anda mungkin memiliki `version: "2"` konfigurasi pipeline, dan OpenSearch Ingestion awalnya

menyediakan pipeline dengan versi 2.6.0. Saat dukungan untuk versi 2.7.0 ditambahkan, dan Anda membuat perubahan pada konfigurasi pipeline, OpenSearch Ingestion memutakhirkan pipeline ke versi 2.7.0. Proses ini membuat pipeline Anda tetap up to date dengan perbaikan bug terbaru dan peningkatan kinerja. OpenSearch Ingestion tidak dapat memperbarui versi utama pipeline Anda kecuali Anda mengubah `version` opsi secara manual dalam konfigurasi pipeline. Untuk informasi selengkapnya, lihat the section called “Memperbarui jaringan pipa”.

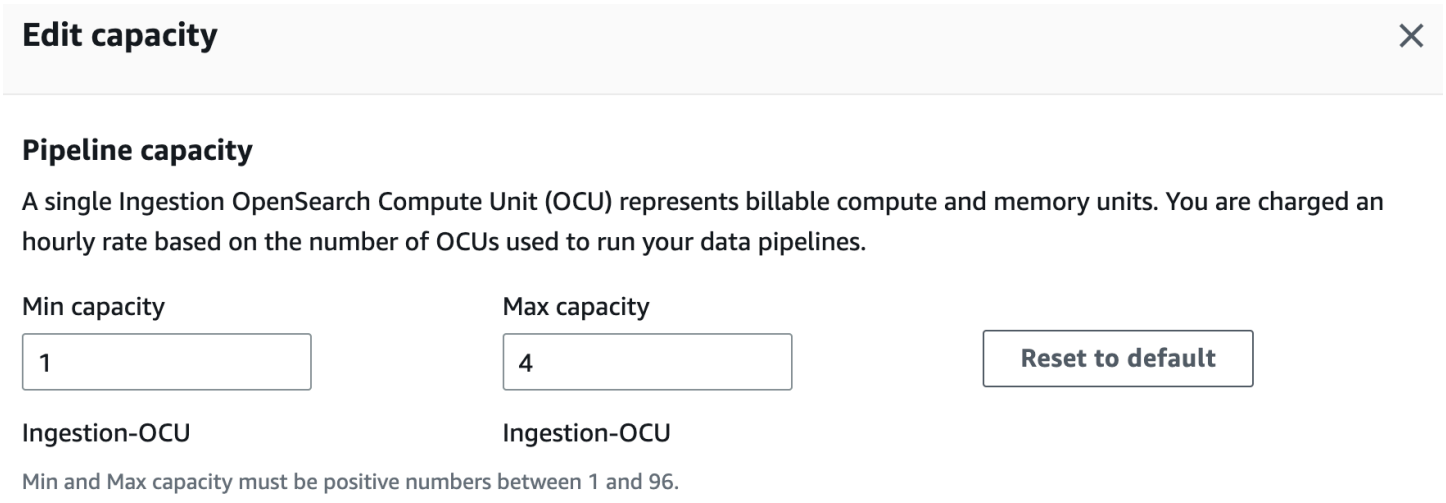
Penskalaan pipa

Anda tidak perlu menyediakan dan mengelola kapasitas pipa sendiri. OpenSearch Penyerapan secara otomatis menskalakan kapasitas pipa Anda sesuai dengan perkiraan beban kerja Anda, berdasarkan Unit OpenSearch Komputasi Tertelan minimum dan maksimum (OCU Ingestion) yang Anda tentukan.

Setiap OCU Ingestion adalah kombinasi dari sekitar 8 GiB memori dan 2 vCPU. Anda dapat menentukan nilai OCU minimum dan maksimum untuk pipeline, dan OpenSearch Ingestion secara otomatis menskalakan kapasitas pipa Anda berdasarkan batas-batas ini.

Anda dapat menentukan salah satu nilai berikut:

- Kapasitas minimum - Pipa dapat mengurangi kapasitas hingga jumlah OCU Tertelan ini. Kapasitas minimum yang ditentukan juga merupakan kapasitas awal untuk pipa.
- Kapasitas maksimum - Pipa dapat meningkatkan kapasitas hingga jumlah OCU Tertelan ini.



Pastikan kapasitas maksimum untuk pipa cukup tinggi untuk menangani lonjakan beban kerja, dan kapasitas minimum cukup rendah untuk meminimalkan biaya saat pipa tidak sibuk. Berdasarkan

pengaturan Anda, OpenSearch Ingestion secara otomatis menskalakan jumlah OCU Ingestion untuk pipeline Anda guna memproses beban kerja yang tertelan. Pada waktu tertentu, Anda hanya dikenakan biaya untuk OCU Ingestion yang sedang digunakan secara aktif oleh pipeline Anda.

Kapasitas yang dialokasikan untuk pipa OpenSearch Ingestion Anda naik turun berdasarkan persyaratan pemrosesan pipa Anda dan beban yang dihasilkan oleh aplikasi klien Anda. Ketika kapasitas dibatasi, OpenSearch Ingestion meningkat dengan mengalokasikan lebih banyak unit komputasi (GiB memori). Saat pipeline Anda memproses beban kerja yang lebih kecil, atau tidak memproses data sama sekali, pipeline dapat menurunkan skala ke OCU Ingestion minimum yang dikonfigurasi.

Anda dapat menentukan minimal 1 OCU Tertelan, maksimum 96 OCU Tertelan untuk jaringan pipa stateless, dan maksimum 48 OCU Tertelan untuk jaringan pipa stateful. Kami merekomendasikan minimal 2 OCU Tertelan untuk sumber berbasis push. Saat buffering persisten diaktifkan, Anda dapat menentukan minimal 2 dan maksimum 384 OCU Ingestion.

Diberikan pipa log standar dengan satu sumber, pola grok sederhana, dan wastafel, setiap unit komputasi dapat mendukung hingga 2 MiB per detik. Untuk jaringan pipa log yang lebih kompleks dengan beberapa prosesor, setiap unit komputasi mungkin mendukung lebih sedikit beban konsumsi. Berdasarkan kapasitas pipa dan pemanfaatan sumber daya, proses penskalaan OpenSearch Ingestion dimulai.

Untuk memastikan ketersediaan yang tinggi, OCU Ingestion didistribusikan di seluruh Availability Zones (AZ). Jumlah AZ tergantung pada kapasitas minimum yang Anda tentukan.

Misalnya, jika Anda menentukan minimal 2 unit komputasi, OCU Ingestion yang digunakan pada waktu tertentu didistribusikan secara merata di 2 AZ. Jika Anda menentukan minimal 3 atau lebih unit komputasi, OCU Ingestion didistribusikan secara merata di 3 AZ. Kami menyarankan Anda menyediakan setidaknya dua OCU Ingestion untuk memastikan ketersediaan 99,9% untuk saluran pipa konsumsi Anda.

Anda tidak ditagih untuk OCU Ingestion saat pipeline berada di`Create failed`,,`Creating`, `Deleting` dan status. `Stopped`

Untuk petunjuk mengkonfigurasi dan mengambil pengaturan kapasitas untuk pipeline, lihatthe section called “Membuat jaringan pipa”.

OpenSearch Harga konsumsi

Pada waktu tertentu, Anda hanya membayar jumlah OCU Tertelan yang dialokasikan ke pipa, terlepas dari apakah ada data yang mengalir melalui pipa. OpenSearch Ingestion segera mengakomodasi beban kerja Anda dengan menskalakan kapasitas pipa naik atau turun berdasarkan penggunaan.

Untuk detail harga selengkapnya, lihat harga OpenSearch Layanan Amazon.

Didukung Wilayah AWS

OpenSearch Konsumsi tersedia dalam subset dari OpenSearch Layanan Wilayah AWS yang tersedia di. Untuk daftar Wilayah yang didukung, lihat titik akhir dan kuota OpenSearch Layanan Amazon di. Referensi Umum AWS

OpenSearch Kuota konsumsi

Untuk daftar kuota default untuk sumber daya OpenSearch Ingestion, lihat Kuota Layanan Amazon OpenSearch .

Menyiapkan peran dan pengguna di Amazon OpenSearch Ingestion

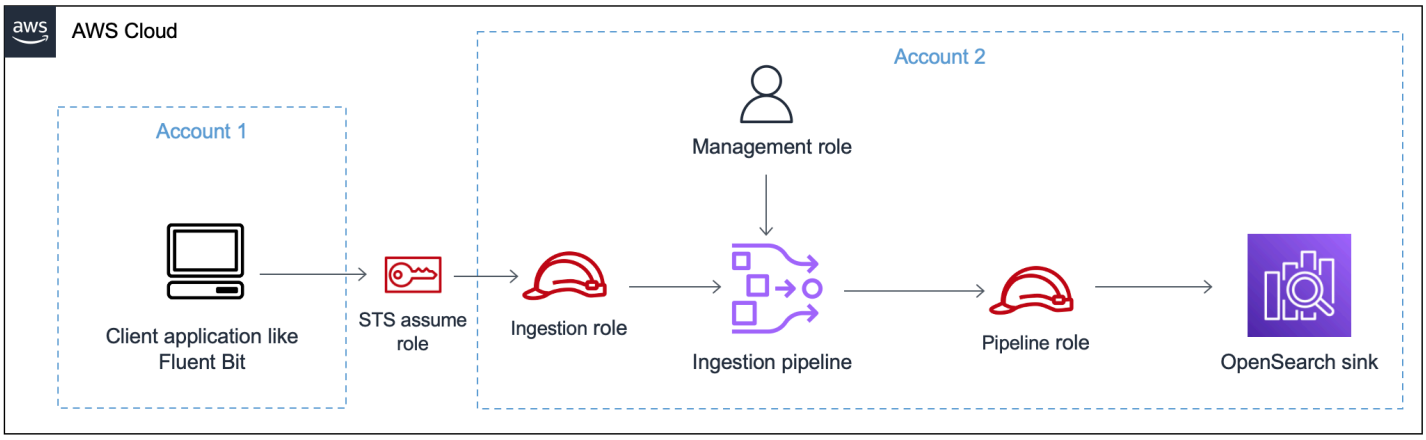
Amazon OpenSearch Ingestion menggunakan berbagai model izin dan peran IAM untuk memungkinkan aplikasi sumber menulis ke saluran pipa, dan untuk memungkinkan saluran pipa menulis ke sink. Sebelum Anda dapat mulai menelan data, Anda perlu membuat satu atau beberapa peran IAM dengan izin tertentu berdasarkan kasus penggunaan Anda.

Minimal, peran berikut diperlukan untuk menyiapkan pipa yang berhasil.

Nama	Penjelasan
Peran manajemen	Setiap prinsipal yang mengelola saluran pipa (umumnya “admin pipa”) memerlukan akses manajemen, yang mencakup izin seperti `osis:CreatePipeline` dan. `osis:UpdatePipeline` Izin ini memungkinkan pengguna untuk mengelola saluran pipa tetapi tidak harus menulis data kepada mereka.

Nama	Penjelasan
Peran pipa	Peran pipeline, yang Anda tentukan dalam konfigurasi YAMAL pipeline, memberikan izin yang diperlukan untuk pipeline untuk menulis ke domain atau sink koleksi dan membaca dari sumber berbasis tarik. Untuk informasi lain, lihat topik berikut: • the section called “Memberikan akses jaringan pipa ke domain” • the section called “Memberikan akses jaringan pipa ke koleksi”
Peran konsumsi	Peran konsumsi berisi `osis:Ingest` izin untuk sumber daya pipa. Izin ini memungkinkan sumber berbasis push untuk menyerap data ke dalam pipeline.

Gambar berikut menunjukkan penyiapan pipeline tipikal, di mana sumber data seperti Amazon S3 atau Fluent Bit menulis ke pipeline di akun yang berbeda. Dalam hal ini, klien perlu mengambil peran konsumsi untuk mengakses pipa. Untuk informasi selengkapnya, lihat the section called “Konsumsi lintas akun”.



Untuk panduan pengaturan sederhana, lihatthe section called “Tutorial: Menelan data ke dalam domain”.

Topik

- the section called “Peran manajemen”
- the section called “Peran konsumsi”
- the section called “Peran pipa”

- the section called “Konsumsi lintas akun”

Peran manajemen

Selain `osis:*` izin dasar yang diperlukan untuk membuat dan memodifikasi pipeline, Anda juga memerlukan `iam:PassRole` izin untuk sumber daya peran pipeline. Setiap Layanan AWS yang menerima peran harus menggunakan izin ini. OpenSearch Ingestion mengasumsikan peran setiap kali perlu menulis data ke wastafel. Ini membantu administrator memastikan bahwa hanya pengguna yang disetujui yang dapat mengonfigurasi OpenSearch Ingestion dengan peran yang memberikan izin. Untuk informasi selengkapnya, lihat Memberikan izin pengguna untuk meneruskan peran ke peran. Layanan AWS

Jika Anda menggunakan AWS Management Console (menggunakan cetak biru dan kemudian memeriksa pipeline Anda), Anda memerlukan izin berikut untuk membuat dan memperbarui pipeline:

```
{
   "Version":"2012-10-17",
   "Statement":[
      {
         "Effect":"Allow",
         "Resource":"*",
         "Action":[
            "osis:CreatePipeline",
            "osis:GetPipelineBlueprint",
            "osis:ListPipelineBlueprints",
            "osis:GetPipeline",
            "osis:ListPipelines",
            "osis:GetPipelineChangeProgress",
            "osis:ValidatePipeline",
            "osis:UpdatePipeline"
         ]
      },
      {
         "Resource":[
            "arn:aws:iam::{your-account-id}:role/pipeline-role"
         ],
         "Effect":"Allow",
         "Action":[
            "iam:PassRole"
         ]
      }
```

```
   ]
}
```

Jika Anda menggunakan AWS CLI (tidak mem-prevalidasi pipeline atau menggunakan cetak biru), Anda memerlukan izin berikut untuk membuat dan memperbarui pipeline:

```
{
   "Version":"2012-10-17",
   "Statement":[
      {
         "Effect":"Allow",
         "Resource":"*",
         "Action":[
            "osis:CreatePipeline",
            "osis:UpdatePipeline"
         ]
      },
      {
         "Resource":[
            "arn:aws:iam::{your-account-id}:role/pipeline-role"
         ],
         "Effect":"Allow",
         "Action":[
            "iam:PassRole"
         ]
      }
   ]
}
```

Peran pipa

Pipa membutuhkan izin tertentu untuk menulis ke wastafelnya. Izin ini bergantung pada apakah wastafel adalah domain OpenSearch Layanan atau koleksi Tanpa OpenSearch Server.

Selain itu, pipeline mungkin memerlukan izin untuk menarik dari aplikasi sumber (jika sumbernya adalah plugin berbasis tarik), dan izin untuk menulis ke antrian huruf mati S3, jika dikonfigurasi.

Topik

- Menulis ke wastafel domain
- Menulis ke wastafel koleksi
- Menulis ke antrian surat mati

Menulis ke wastafel domain

Pipeline OpenSearch Ingestion memerlukan izin untuk menulis ke domain OpenSearch Layanan yang dikonfigurasi sebagai wastafelnya. Izin ini mencakup kemampuan untuk mendeskripsikan domain dan mengirim permintaan HTTP ke sana.

Untuk menyediakan pipeline Anda dengan izin yang diperlukan untuk menulis ke wastafel, pertama-tama buat peran AWS Identity and Access Management (IAM) dengan izin yang diperlukan. Izin ini sama untuk saluran pipa publik dan VPC. Kemudian, tentukan peran pipeline dalam kebijakan akses domain sehingga domain dapat menerima permintaan tulis dari pipeline.

Terakhir, tentukan peran ARN sebagai nilai opsi sts_role_arn dalam konfigurasi pipeline:

```
version: "2"
source:
  http:
    ...
processor:
  ...
sink:
  - opensearch:
      ...
      aws:
        sts_role_arn: arn:aws:iam::{your-account-id}:role/pipeline-role
```

Untuk petunjuk untuk menyelesaikan setiap langkah ini, lihat Mengizinkan saluran pipa mengakses domain.

Menulis ke wastafel koleksi

Pipeline OpenSearch Ingestion memerlukan izin untuk menulis ke koleksi OpenSearch Tanpa Server yang dikonfigurasi sebagai wastafelnya. Izin ini mencakup kemampuan untuk mendeskripsikan koleksi dan mengirim permintaan HTTP ke dalamnya.

Pertama, buat peran IAM yang memiliki `aoss:BatchGetCollection` izin terhadap semua sumber daya (`*`). Kemudian, sertakan peran ini dalam kebijakan akses data dan berikan izin untuk membuat indeks, memperbarui indeks, mendeskripsikan indeks, dan menulis dokumen dalam koleksi. Terakhir, tentukan peran ARN sebagai nilai opsi sts_role_arn dalam konfigurasi pipeline.

Untuk petunjuk untuk menyelesaikan setiap langkah ini, lihat Mengizinkan saluran pipa mengakses koleksi.

Menulis ke antrian surat mati

Jika Anda mengonfigurasi pipeline untuk menulis ke antrian huruf mati (DLQ), Anda harus menyertakan `sts_role_arn` opsi dalam konfigurasi DLQ. Izin yang disertakan dalam peran ini memungkinkan pipeline mengakses bucket S3 yang Anda tentukan sebagai tujuan untuk peristiwa DLQ.

Anda harus menggunakan yang sama `sts_role_arn` di semua komponen pipa. Oleh karena itu, Anda harus melampirkan kebijakan izin terpisah ke peran pipeline yang menyediakan akses DLQ. Minimal, peran harus diizinkan `S3:PutObject` tindakan pada sumber daya bucket:

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Sid": "WriteToS3DLQ",
      "Effect": "Allow",
      "Action": "s3:PutObject",
      "Resource": "arn:aws:s3:::my-dlq-bucket/*"
    }
  ]
}
```

Anda kemudian dapat menentukan peran dalam konfigurasi DLQ pipeline:

```
  ...
  sink:
    opensearch:
      dlq:
        s3:
          bucket: "my-dlq-bucket"
          key_path_prefix: "dlq-files"
          region: "us-west-2"
          sts_role_arn: "arn:aws:iam::123456789012:role/pipeline-role"
```

Peran konsumsi

Semua plugin sumber yang saat ini didukung oleh OpenSearch Ingestion, dengan pengecualian S3, menggunakan arsitektur berbasis push. Ini berarti bahwa aplikasi sumber mendorong data ke pipa, daripada pipa yang menarik data dari sumbernya.

Oleh karena itu, Anda harus memberikan izin yang diperlukan kepada aplikasi sumber Anda untuk menyerap data ke dalam pipeline OpenSearch Ingestion. Minimal, peran yang menandatangani permintaan harus diberikan izin untuk `osis:Ingest` tindakan tersebut, yang memungkinkannya mengirim data ke pipeline. Izin yang sama diperlukan untuk titik akhir saluran pipa publik dan VPC.

Contoh kebijakan berikut memungkinkan prinsipal terkait untuk menyerap data ke dalam satu pipeline yang disebut`my-pipeline`:

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Sid": "PermitsWriteAccessToPipeline",
      "Effect": "Allow",
      "Action": "osis:Ingest",
      "Resource": "arn:aws:osis:us-west-2:{your-account-id}:pipeline/my-pipeline"
    }
  ]
}
```

Untuk informasi selengkapnya, lihat the section called “Bekerja dengan integrasi pipa”.

Konsumsi lintas akun

Anda mungkin perlu memasukkan data ke dalam pipeline dari yang berbedaAkun AWS, seperti akun aplikasi. Untuk mengonfigurasi konsumsi lintas akun, tentukan peran konsumsi dalam akun yang sama dengan pipeline dan buat hubungan kepercayaan antara peran konsumsi dan akun aplikasi:

```
{
  "Version": "2012-10-17",
  "Statement": [{
     "Effect": "Allow",
     "Principal": {
       "AWS": "arn:aws:iam::{external-account-id}:root"
      },
     "Action": "sts:AssumeRole"
  }]
}
```

Kemudian, konfigurasikan aplikasi Anda untuk mengambil peran konsumsi. Akun aplikasi harus memberikan AssumeRoleizin peran aplikasi untuk peran konsumsi di akun pipeline.

Untuk langkah-langkah rinci dan contoh kebijakan IAM, lihatthe section called “Menyediakan akses konsumsi lintas akun”.

Memberikan akses saluran pipa Amazon OpenSearch Ingestion ke domain

Pipeline Amazon OpenSearch Ingestion memerlukan izin untuk menulis ke domain OpenSearch Layanan yang dikonfigurasi sebagai wastafelnya. Untuk menyediakan akses, Anda mengonfigurasi peran AWS Identity and Access Management (IAM) dengan kebijakan izin terbatas yang membatasi akses ke domain tempat pipeline mengirim data. Misalnya, Anda mungkin ingin membatasi pipeline konsumsi hanya pada domain dan indeks yang diperlukan untuk mendukung kasus penggunaannya.

Sebelum menentukan peran dalam konfigurasi pipeline, Anda harus mengonfigurasinya dengan hubungan kepercayaan yang sesuai, lalu memberinya akses ke domain dalam kebijakan akses domain.

Topik

- Langkah 1: Buat peran pipeline
- Langkah 2: Sertakan peran pipeline dalam kebijakan akses domain
- Langkah 3: Petakan peran pipeline (hanya untuk domain yang menggunakan kontrol akses berbutir halus)
- Langkah 4: Tentukan peran dalam konfigurasi pipeline

Langkah 1: Buat peran pipeline

Peran yang Anda tentukan dalam parameter sts_role_arn dari konfigurasi pipeline harus memiliki kebijakan izin terlampir yang memungkinkannya mengirim data ke sink domain. Itu juga harus memiliki hubungan kepercayaan yang memungkinkan OpenSearch Ingestion untuk mengambil peran. Untuk petunjuk tentang cara melampirkan kebijakan ke peran, lihat Menambahkan izin identitas IAM di Panduan Pengguna IAM.

Kebijakan contoh berikut menunjukkan hak istimewa paling sedikit yang dapat Anda berikan dalam peran sts_role_arn konfigurasi pipeline agar dapat ditulis ke satu domain:

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

```
            "Action": "es:DescribeDomain",
            "Resource": "arn:aws:es:*:{your-account-id}:domain/*"
        },
        {
            "Effect": "Allow",
            "Action": "es:ESHttp*",
            "Resource": "arn:aws:es:*:{your-account-id}:domain/{domain-namedomain}/*"
        }
    ]
}
```

Jika Anda berencana untuk menggunakan kembali peran untuk menulis ke beberapa domain, Anda dapat membuat kebijakan lebih luas dengan mengganti nama domain dengan karakter wildcard (). *

Peran tersebut harus memiliki hubungan kepercayaan berikut, yang memungkinkan OpenSearch Ingestion untuk mengambil peran pipeline:

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

Selain itu, kami menyarankan Anda menambahkan kunci `aws:SourceAccount` dan `aws:SourceArn` kondisi ke kebijakan untuk melindungi diri Anda dari masalah wakil yang membingungkan. Akun sumber adalah pemilik pipa.

Misalnya, Anda dapat menambahkan blok kondisi berikut ke kebijakan:

```
"Condition": {
    "StringEquals": {
        "aws:SourceAccount": "{your-account-id}"
    },
    "ArnLike": {
        "aws:SourceArn": "arn:aws:osis:{region}:{your-account-id}:pipeline/*"
```

```
    }
}
```

Langkah 2: Sertakan peran pipeline dalam kebijakan akses domain

Agar pipeline dapat menulis data ke domain, domain harus memiliki kebijakan akses tingkat domain yang memungkinkan peran pipeline sts_role_arn untuk mengaksesnya.

Contoh kebijakan akses domain berikut memungkinkan peran pipeline bernama`pipeline-role`, yang Anda buat pada langkah sebelumnya, untuk menulis data ke domain bernama`ingestion-domain`:

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Effect": "Allow",
      "Principal": {
        "AWS": "arn:aws:iam::{your-account-id}:role/{pipeline-role}"
      },
      "Action": ["es:DescribeDomain", "es:ESHttp*"],
      "Resource": "arn:aws:es:{region}:{your-account-id}:domain/{domain-name}/*"
    }
  ]
}
```

Langkah 3: Petakan peran pipeline (hanya untuk domain yang menggunakan kontrol akses berbutir halus)

Jika domain Anda menggunakan kontrol akses berbutir halus untuk autentikasi, ada langkah-langkah tambahan yang perlu Anda ambil untuk menyediakan akses pipeline Anda ke domain. Langkah-langkahnya berbeda tergantung pada konfigurasi domain Anda:

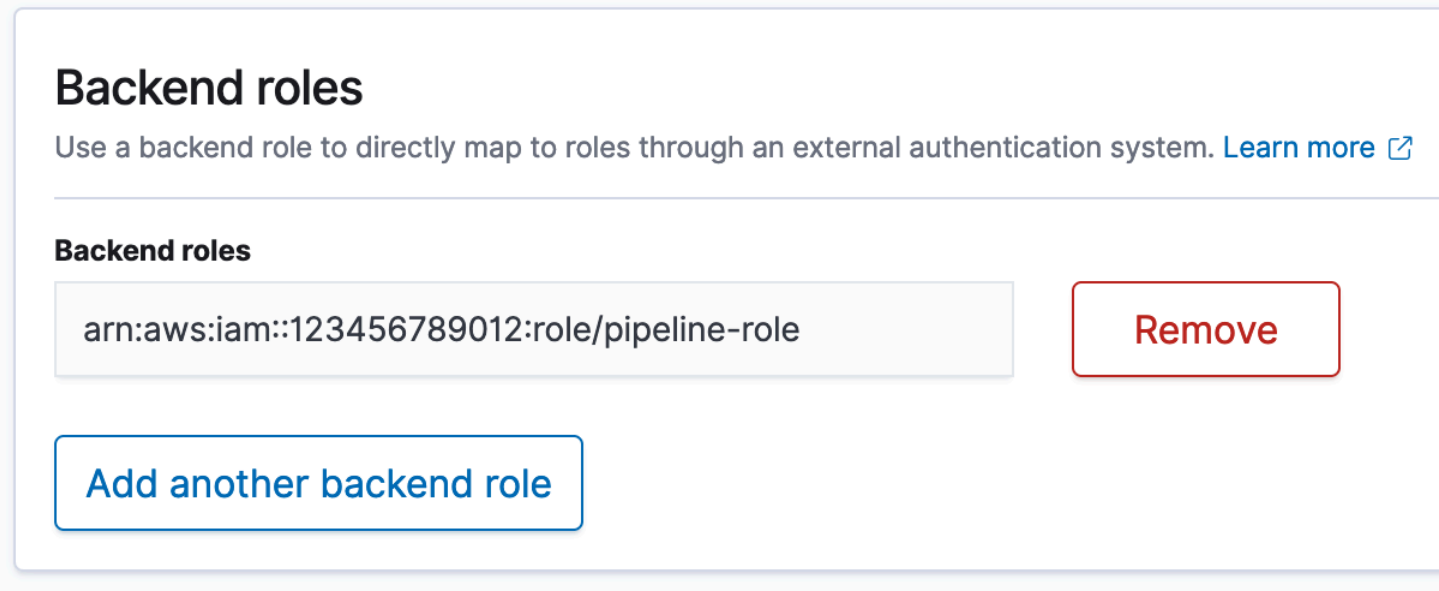
Skenario 1: Peran master dan peran pipeline yang berbeda — Jika Anda menggunakan Nama Sumber Daya Amazon IAM (ARN) sebagai pengguna master dan ini berbeda dengan peran pipeline `sts_role_arn` (), Anda perlu memetakan peran pipeline ke OpenSearch `all_access` peran backend. Ini pada dasarnya menambahkan peran pipeline sebagai pengguna master tambahan. Untuk informasi selengkapnya, lihat Pengguna master tambahan.

Skenario 2: Menguasai pengguna dalam database pengguna internal — Jika domain Anda menggunakan pengguna master di database pengguna internal dan otentikasi dasar HTTP untuk

OpenSearch Dasbor, Anda tidak dapat meneruskan nama pengguna dan kata sandi utama langsung ke konfigurasi pipeline. Sebagai gantinya, Anda perlu memetakan peran pipeline (`sts_role_arn`) ke peran OpenSearch `all_access` backend. Ini pada dasarnya menambahkan peran pipeline sebagai pengguna master tambahan. Untuk informasi selengkapnya, lihat Pengguna master tambahan.

Skenario 3: Peran master dan peran pipeline yang sama (tidak umum) - Jika Anda menggunakan ARN IAM sebagai pengguna master, dan itu adalah ARN yang sama yang Anda gunakan sebagai peran pipeline (`sts_role_arn`), Anda tidak perlu mengambil tindakan lebih lanjut. Pipeline memiliki izin yang diperlukan untuk menulis ke domain. Skenario ini jarang terjadi karena sebagian besar lingkungan menggunakan peran admin atau peran lain sebagai peran utama.

Gambar berikut menunjukkan cara memetakan peran pipeline ke peran backend:



Langkah 4: Tentukan peran dalam konfigurasi pipeline

Agar berhasil membuat pipeline, Anda harus menentukan peran pipeline yang Anda buat di langkah 1 sebagai parameter sts_role_arn dalam konfigurasi pipeline Anda. Pipeline mengasumsikan peran ini untuk menandatangani permintaan ke sink domain OpenSearch Layanan.

Di `sts_role_arn` lapangan, tentukan ARN dari peran pipa IAM:

```
version: "2"
log-pipeline:
  source:
    http:
      path: "/${pipelineName}/logs"
```

```
processor:
  - grok:
      match:
        log: [ "%{COMMONAPACHELOG}" ]
sink:
  - opensearch:
      hosts: [ "https://search-{domain-name}.us-east-1.es.amazonaws.com" ]
      index: "my-index"
      aws:
        region: "[region}"
        sts_role_arn: "arn:aws:iam::{your-account-id}:role/{pipeline-role}"
```

Untuk referensi lengkap parameter yang diperlukan dan tidak didukung, lihatthe section called “Plugin dan opsi yang didukung”.

Memberikan akses saluran pipa Amazon OpenSearch Ingestion ke koleksi

Pipeline Amazon OpenSearch Ingestion memerlukan izin untuk menulis ke koleksi OpenSearch Tanpa Server yang dikonfigurasi sebagai wastafel. Untuk menyediakan akses, Anda mengonfigurasi peran AWS Identity and Access Management (IAM) dengan kebijakan izin terbatas yang membatasi akses ke koleksi tempat pipeline mengirim data. OpenSearch Konsumsi dapat menelan data ke koleksi publik dan koleksi VPC.

Sebelum menentukan peran dalam konfigurasi pipeline, Anda harus mengonfigurasinya dengan hubungan kepercayaan yang sesuai, lalu memberinya izin akses data ke indeks koleksi.

Topik

- Batasan
- Langkah 1: Buat peran pipeline
- Langkah 2: Buat koleksi
- Langkah 3: Buat pipeline

Batasan

Batasan berikut berlaku untuk pipeline yang menulis ke koleksi Tanpa OpenSearch Server:

- Prosesor grup jejak OTel saat ini tidak berfungsi dengan sink OpenSearch koleksi Tanpa Server.
- Saat ini, OpenSearch Ingestion hanya mendukung `_template` operasi lama, sementara OpenSearch Serverless mendukung operasi composable. `_index_template` Oleh

karena itu, jika konfigurasi pipeline Anda menyertakan `index_type` opsi, itu harus diatur kemanagement_disabled.

Langkah 1: Buat peran pipeline

Peran yang Anda tentukan dalam parameter sts_role_arn dari konfigurasi pipeline harus memiliki kebijakan izin terlampir yang memungkinkannya mengirim data ke sink koleksi. Itu juga harus memiliki hubungan kepercayaan yang memungkinkan OpenSearch Ingestion untuk mengambil peran. Untuk petunjuk tentang cara melampirkan kebijakan ke peran, lihat Menambahkan izin identitas IAM di Panduan Pengguna IAM.

Kebijakan contoh berikut menunjukkan hak istimewa paling sedikit yang dapat Anda berikan dalam peran sts_role_arn konfigurasi pipeline agar dapat ditulis ke koleksi:

```
{
    "Version": "2012-10-17",
    "Statement": [
        {
            "Action": [
                "aoss:BatchGetCollection"
            ],
            "Effect": "Allow",
            "Resource": "arn:aws:aoss:{region}:{your-account-
id}:collection/{collection-id}"
        },
        {
            "Action": [
                "aoss:CreateSecurityPolicy",
                "aoss:GetSecurityPolicy",
                "aoss:UpdateSecurityPolicy",
                "aoss:APIAccessAll"
            ],
            "Effect": "Allow",
            "Resource": "*",
            "Condition": {
                "StringEquals": {
                    "aoss:collection": "{collection-name}"
                }
            }
        }
    ]
}
```

Peran tersebut harus memiliki hubungan kepercayaan berikut, yang memungkinkan OpenSearch Ingestion untuk menganggapnya:

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

Selain itu, kami menyarankan Anda menambahkan kunci `aws:SourceAccount` dan `aws:SourceArn` kondisi ke kebijakan untuk melindungi diri Anda dari masalah wakil yang membingungkan. Akun sumber adalah pemilik pipa.

Misalnya, Anda dapat menambahkan blok kondisi berikut ke kebijakan:

```
"Condition": {
    "StringEquals": {
        "aws:SourceAccount": "{your-account-id}"
    },
    "ArnLike": {
        "aws:SourceArn": "arn:aws:osis:{region}:{your-account-id}:pipeline/*"
    }
}
```

Langkah 2: Buat koleksi

Buat koleksi OpenSearch Tanpa Server dengan pengaturan berikut:

- Kebijakan akses data berikut, yang memberikan izin yang diperlukan untuk peran pipeline:

```
[
  {
```

```
    "Rules": [
      {
        "Resource": [
          "index/{collection-name}/*"
        ],
        "Permission": [
          "aoss:CreateIndex",
          "aoss:UpdateIndex",
          "aoss:DescribeIndex",
          "aoss:WriteDocument",
        ],
        "ResourceType": "index"
      }
    ],
    "Principal": [
      "arn:aws:iam::{account-id}:role/{pipeline-role}"
    ],
    "Description": "Pipeline role access"
  }
]
```

> **Note**
>
> Dalam `Principal` elemen, tentukan Nama Sumber Daya Amazon (ARN) dari peran pipeline yang Anda buat di langkah sebelumnya.

- Kebijakan akses jaringan. Anda dapat menyerap data ke dalam koleksi publik atau koleksi VPC. Jika Anda menggunakan koleksi VPC, kebijakan jaringan harus mengizinkan satu atau beberapa titik akhir VPC untuk mengakses koleksi. Misalnya, Anda dapat menambahkan kebijakan jaringan berikut, yang memungkinkan satu titik akhir VPC mengakses koleksi:

```
[
   {
      "Description":"VPC access",
      "Rules":[
         {
            "ResourceType":"collection",
            "Resource":[
               "collection/{collection-name}"
            ]
         }
      ],
```

```
      "AllowFromPublic": false,
      "SourceVPCEs":[
         "vpce-050f79086ee71ac05"
      ]
   }
]
```

> **Note**
>
> Selain itu, Anda harus menentukan nama kebijakan jaringan dalam `network_policy_name` opsi dalam konfigurasi pipeline. Lihat langkah 3 untuk contoh konfigurasi pipeline.

Untuk instruksi untuk membuat koleksi, lihatthe section called “Membuat koleksi”.

Langkah 3: Buat pipeline

Terakhir, buat pipeline tempat Anda menentukan peran pipeline dan detail koleksi. Pipeline mengasumsikan peran ini untuk menandatangani permintaan ke sink koleksi OpenSearch Tanpa Server.

Pastikan untuk melakukan hal berikut:

- Untuk `hosts` opsi, tentukan titik akhir koleksi yang Anda buat di langkah 2.
- Untuk `sts_role_arn` opsi, tentukan Nama Sumber Daya Amazon (ARN) dari peran pipeline yang Anda buat di langkah 1.
- Atur `serverless` opsi ke`true`.
- Tetapkan `network_policy_name` opsi ke nama kebijakan jaringan yang dilampirkan pada koleksi.

```
version: "2"
log-pipeline:
  source:
    http:
        path: "/log/ingest"
  processor:
    - date:
        from_time_received: true
```

```
        destination: "@timestamp"
  sink:
    - opensearch:
        hosts: [ "https://{collection-id}.{region}.aoss.amazonaws.com" ]
        index: "my-index"
        aws:
          serverless: true
          serverless_options:
            network_policy_name: "{network-policy-name}" # If the policy doesn't exist,
a new policy is created.
          region: "us-east-1"
          sts_role_arn: "arn:aws:iam::{account-id}:role/{pipeline-role}"
```

Untuk referensi lengkap parameter yang diperlukan dan tidak didukung, lihatthe section called “Plugin dan opsi yang didukung”.

Memulai dengan Amazon OpenSearch Ingestion

Amazon OpenSearch Ingestion mendukung penyerapan data ke dalam domain OpenSearch Layanan terkelola dan koleksi Tanpa Server. OpenSearch Tutorial berikut memandu Anda melalui langkah-langkah dasar untuk mendapatkan pipa dan berjalan untuk setiap kasus penggunaan ini.

> **Note**
>
> Pembuatan pipeline akan gagal jika Anda tidak mengatur izin yang benar. Lihat pemahaman the section called “Menyiapkan peran dan pengguna” yang lebih baik tentang peran yang diperlukan sebelum Anda membuat pipeline.

Topik

- Tutorial: Menelan data ke dalam domain menggunakan Amazon Ingestion OpenSearch
- Tutorial: Menelan data ke dalam koleksi menggunakan Amazon OpenSearch Ingestion

Tutorial: Menelan data ke dalam domain menggunakan Amazon Ingestion OpenSearch

Tutorial ini menunjukkan Anda bagaimana menggunakan Amazon OpenSearch Ingestion untuk mengonfigurasi alur sederhana dan mengonfigurasi data ke domain Amazon OpenSearch Service.

Sebuah pipa adalah sumber daya yang OpenSearch menelan ketentuan dan mengelola. Anda dapat menggunakan pipeline untuk memfilter, memperkaya, mengubah, menormalkan, dan mengumpulkan data untuk analisis hilir dan visualisasi dalam Layanan. OpenSearch

Tutorial ini memandu Anda melalui langkah-langkah dasar untuk mendapatkan alur dan berjalan dengan cepat. Untuk informasi lebih terperinci, lihat the section called “Membuat jaringan pipa”.

Anda akan menyelesaikan langkah-langkah berikut dalam tutorial ini:

1. Buat peran pipeline.
2. Buat domain.
3. Buat pipeline.
4. Menelan beberapa data sampel.

Dalam tutorial ini, Anda akan membuat sumber daya berikut:

- Sebuah pipa bernama `ingestion-pipeline`
- Sebuah domain bernama `ingestion-domain` bahwa pipeline akan menulis ke
- Peran IAM bernama `PipelineRole` bahwa pipeline akan berasumsi untuk menulis ke domain

Izin yang diperlukan

Untuk menyelesaikan tutorial ini, Anda harus memiliki izin IAM yang benar. Pengguna atau peran Anda harus memiliki kebijakan berbasis identitas terlampir dengan izin minimum berikut. Izin ini memungkinkan Anda untuk membuat peran pipeline (`iam:Create`), membuat atau memodifikasi domain (`es:*`), dan bekerja dengan pipeline (`osis:*`).

Selain itu, `iam:PassRole` izin diperlukan pada sumber daya peran pipa. Izin ini memungkinkan Anda untuk meneruskan peran pipeline ke OpenSearch Penyerapan sehingga dapat menulis data ke domain.

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

```
            "osis:*",
            "iam:Create*",
            "es:*"
         ]
      },
      {
         "Resource":[
            "arn:aws:iam::{your-account-id}:role/PipelineRole"
         ],
         "Effect":"Allow",
         "Action":[
            "iam:PassRole"
         ]
      }
   ]
}
```

Langkah 1: Buat peran alur

Pertama, buat peran yang akan diasumsikan pipeline untuk mengakses wastafel domain OpenSearch Layanan. Anda akan menyertakan peran ini dalam konfigurasi pipeline nanti dalam tutorial ini.

Untuk membuat peran alur

1. Buka AWS Identity and Access Management konsol di https://console.aws.amazon.com/iamv2/.
2. Pilih Kebijakan, lalu pilih Buat kebijakan.
3. Dalam tutorial ini, Anda akan menelan data ke dalam domain yang disebut`ingestion-domain`, yang akan Anda buat pada langkah berikutnya. Pilih JSON dan tempelkan kebijakan berikut ke dalam editor. Ganti `{your-account-id}` dengan ID akun Anda, dan ubah Wilayah jika perlu.

   ```
   {
       "Version": "2012-10-17",
       "Statement": [
           {
               "Effect": "Allow",
               "Action": "es:DescribeDomain",
               "Resource": "arn:aws:es:us-east-1:{your-account-id}:domain/ingestion-
   domain"
           },
           {
               "Effect": "Allow",
   ```

```
            "Action": "es:ESHttp*",
            "Resource": "arn:aws:es:us-east-1:{your-account-id}:domain/ingestion-
domain/*"
        }
    ]
}
```

Jika Anda ingin menulis data ke domain yang ada, ganti `ingestion-domain` dengan nama domain Anda.

> **Note**
>
> Untuk kesederhanaan dalam tutorial ini, kami menggunakan kebijakan akses yang cukup luas. Namun, dalam lingkungan produksi, kami menyarankan Anda menerapkan kebijakan akses yang lebih ketat ke peran pipeline Anda. Untuk contoh kebijakan yang menyediakan izin minimum yang diperlukan, lihatthe section called “Memberikan akses jaringan pipa ke domain”.

4. Pilih Berikutnya, pilih Berikutnya, dan beri nama kebijakan pipeline kebijakan Anda.
5. Pilih Buat kebijakan.
6. Selanjutnya, buat peran dan lampirkan kebijakan ke dalamnya. Pilih Peran, lalu pilih Buat peran.
7. Pilih Kebijakan kepercayaan khusus dan tempelkan kebijakan berikut ke editor:

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

8. Pilih Selanjutnya. Kemudian cari dan pilih pipeline-policy (yang baru saja Anda buat).
9. Pilih Berikutnya dan beri nama peran PipelineRole.
10. Pilih Create role (Buat peran).

Ingat Amazon Resource Name (ARN) peran tersebut (misalnya,`arn:aws:iam::{your-account-id}:role/PipelineRole`). Anda akan memerlukannya saat Anda membuat alur Anda.

Langkah 2: Buat domain

Selanjutnya, buat domain bernama `ingestion-domain` untuk menelan data ke dalam.

Arahkan ke konsol OpenSearch Layanan Amazon di https://console.aws.amazon.com/aos/home dan buat domain yang memenuhi persyaratan berikut:

- Sedang berjalan OpenSearch 1.0 atau lebih baru, atau Elasticsearch 7.4 atau lebih baru
- Menggunakan akses publik
- Tidak menggunakan kontrol akses detail

> **Note**
>
> Persyaratan ini dimaksudkan untuk memastikan kesederhanaan dalam tutorial ini. Di lingkungan produksi, Anda dapat mengonfigurasi domain dengan akses VPC dan/atau menggunakan kontrol akses berbutir halus. Untuk instruksi, lihat topik selanjutnya dalam chapter ini.

Domain harus memiliki kebijakan akses yang memberikan izin`PipelineRole`, yang Anda buat pada langkah sebelumnya. Pipeline akan mengasumsikan peran ini (bernama sts_role_arn dalam konfigurasi pipeline) untuk mengirim data ke wastafel domain Service. OpenSearch

Pastikan domain memiliki kebijakan akses tingkat domain berikut, yang memberikan `PipelineRole` akses ke domain. Ganti Wilayah dan ID akun dengan milik Anda sendiri:

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Effect": "Allow",
      "Principal": {
        "AWS": "arn:aws:iam::{your-account-id}:role/PipelineRole"
      },
      "Action": "es:*",
      "Resource": "arn:aws:es:us-east-1:{your-account-id}:domain/ingestion-domain/*"
```

```
    }
  ]
}
```

Untuk informasi selengkapnya tentang membuat kebijakan akses tingkat domain, lihat Kebijakan akses berbasis sumber daya.

Jika Anda sudah memiliki domain yang dibuat, ubah kebijakan akses yang ada untuk memberikan izin di atas. `PipelineRole`

> **Note**
>
> Ingat endpoint domain (misalnya,`https://search-`*`ingestion-domain`*`.us-east-1.es.amazonaws.com`). Anda akan menggunakannya pada langkah berikutnya untuk mengkonfigurasi pipeline Anda.

Langkah 3: Buat Alur

Sekarang setelah Anda memiliki domain dan peran dengan hak akses yang sesuai, Anda dapat membuat alur.

Untuk membuat alur

1. Dalam konsol OpenSearch Layanan Amazon, pilih Pipelines dari panel navigasi kiri.
2. Pilih Buat pipeline.
3. Beri nama pipeline ingestion-pipeline dan pertahankan pengaturan kapasitas sebagai defaultnya.
4. Dalam tutorial ini, Anda akan membuat sub-pipeline sederhana `log-pipeline` yang disebut yang menggunakan plugin sumber Http. Plugin ini menerima data log dalam format array JSON. Anda akan menentukan domain OpenSearch Service tunggal sebagai wastafel, dan menelan semua data ke dalam `application_logs` indeks.

   Di bawah konfigurasi Pipeline, paste konfigurasi YAKL berikut ke editor:

   ```
   version: "2"
   log-pipeline:
     source:
       http:
         path: "/${pipelineName}/test_ingestion_path"
   ```

```
processor:
  - date:
      from_time_received: true
      destination: "@timestamp"
sink:
  - opensearch:
      hosts: [ "https://search-ingestion-domain.us-east-1.es.amazonaws.com" ]
      index: "application_logs"
      aws:
        sts_role_arn: "arn:aws:iam::{your-account-id}:role/PipelineRole"
        region: "us-east-1"
```

> **Note**
>
> `path`Opsi menentukan jalur URI untuk konsumsi. Opsi ini diperlukan untuk sumber berbasis tarik. Untuk informasi selengkapnya, lihat the section called “Menentukan jalur konsumsi”.

5. Ganti `hosts` URL dengan titik akhir domain yang Anda buat (atau modifikasi) di bagian sebelumnya. Ganti `sts_role_arn` parameter dengan ARN dari. `PipelineRole`
6. Pilih Validasi pipeline dan pastikan validasi berhasil.
7. Untuk kesederhanaan dalam tutorial ini, mengonfigurasi akses publik untuk alur. Di bawah Jaringan, pilih Akses publik.

   Untuk informasi tentang mengonfigurasi akses VPC, lihat. the section called “Mengamankan jaringan pipa dalam VPC”
8. Tetap aktifkan penerbitan log jika Anda mengalami masalah saat menyelesaikan tutorial ini. Untuk informasi selengkapnya, lihat the section called “Memantau log”.

   Tentukan nama grup log berikut: `/aws/vendedlogs/OpenSearchIngestion/ingestion-pipeline/audit-logs`
9. Pilih Selanjutnya. Tinjau konfigurasi pipeline Anda dan pilih Buat pipeline. Pipa membutuhkan waktu 5-10 menit untuk menjadi aktif.

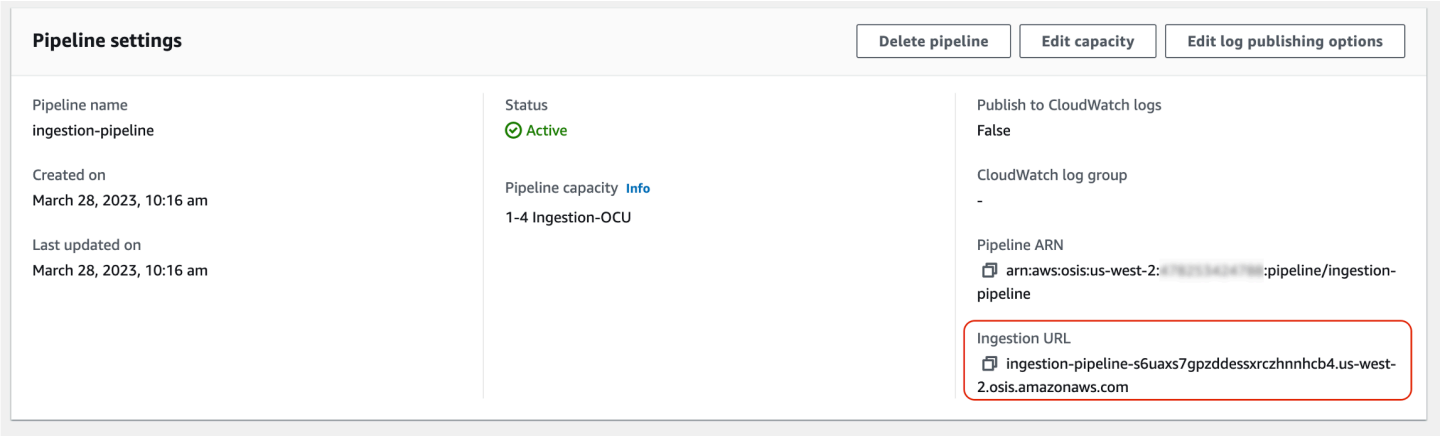
Langkah 4: Menelan beberapa data sampel

Ketika status pipeline`Active`, Anda dapat mulai menelan data ke dalamnya. Anda harus menandatangani semua permintaan HTTP ke pipeline menggunakan Signature Version 4. Gunakan

alat HTTP seperti Postman atau awscurl untuk mengirim beberapa data ke pipeline. Seperti halnya mengindeks data langsung ke domain, menelan data ke dalam pipeline selalu membutuhkan peran IAM atau kunci akses IAM dan kunci rahasia.

> **Note**
>
> Kepala sekolah yang menandatangani permintaan harus memiliki izin `osis:Ingest` IAM.

Pertama, dapatkan URL konsumsi dari halaman pengaturan Pipeline:



Kemudian, menelan beberapa data sampel. Permintaan berikut menggunakan awscurl untuk mengirim file log tunggal ke indeks: `application_logs`

```
awscurl --service osis --region us-east-1 \
    -X POST \
    -H "Content-Type: application/json" \
    -d
 '[{"time":"2014-08-11T11:40:13+00:00","remote_addr":"122.226.223.69","status":"404","request":
 http://www.k2proxy.com//hello.html HTTP/1.1","http_user_agent":"Mozilla/4.0
 (compatible; WOW64; SLCC2;)"}]' \
    https://{pipeline-endpoint}.us-east-1.osis.amazonaws.com/log-pipeline/
test_ingestion_path
```

Anda harus melihat `200 OK` tanggapan. Jika Anda mendapatkan kesalahan autentikasi, itu mungkin karena Anda menelan data dari akun terpisah daripada pipeline. Lihat the section called “Memperbaiki masalah izin”.

Sekarang, kueri `application_logs` indeks untuk memastikan bahwa entri log Anda berhasil dicerna:

```
awscurl --service es --region us-east-1 \
     -X GET \
     https://search-{ingestion-domain}.us-east-1.es.amazonaws.com/application_logs/
_search | json_pp
```

Respon sampel:


```
{
   "took":984,
   "timed_out":false,
   "_shards":{
      "total":1,
      "successful":5,
      "skipped":0,
      "failed":0
   },
   "hits":{
      "total":{
         "value":1,
         "relation":"eq"
      },
      "max_score":1.0,
      "hits":[
         {
            "_index":"application_logs",
            "_type":"_doc",
            "_id":"z6VY_IMBRpceX-DU6V4O",
            "_score":1.0,
            "_source":{
               "time":"2014-08-11T11:40:13+00:00",
               "remote_addr":"122.226.223.69",
               "status":"404",
               "request":"GET http://www.k2proxy.com//hello.html HTTP/1.1",
               "http_user_agent":"Mozilla/4.0 (compatible; WOW64; SLCC2;)",
               "@timestamp":"2022-10-21T21:00:25.502Z"
            }
         }
      ]
   }
}
```

Memperbaiki masalah izin

Jika Anda mengikuti langkah-langkah dalam tutorial dan Anda masih melihat kesalahan otentikasi ketika Anda mencoba menelan data, mungkin karena peran yang menulis ke pipeline berbeda Akun AWS dari pipeline itu sendiri. Dalam hal ini, Anda perlu membuat dan mengambil peran yang secara khusus memungkinkan Anda mengonsumsi data. Untuk petunjuk, lihat the section called "Menyediakan akses konsumsi lintas akun".

Sumber daya terkait

Tutorial ini menyajikan kasus penggunaan sederhana menelan satu dokumen melalui HTTP. Dalam skenario produksi, kamu akan mengonfigurasi aplikasi klien kamu (seperti Fluent Bit, Kubernetes, atau OpenTelemetry Collector) untuk mengirim data ke satu atau beberapa pipeline. Pipelines Anda kemungkinan akan lebih kompleks daripada contoh sederhana dalam tutorial ini.

Untuk mulai mengonfigurasi klien Anda dan menelan data, lihat sumber daya berikut:

- Membuat dan mengelola jaringan pipa
- Mengkonfigurasi klien Anda untuk mengirim data ke OpenSearch Penyerapan
- Dokumentasi Data Prepper

Tutorial: Menelan data ke dalam koleksi menggunakan Amazon OpenSearch Ingestion

Tutorial ini menunjukkan cara menggunakan Amazon OpenSearch Ingestion untuk mengonfigurasi pipeline sederhana dan menyerap data ke dalam koleksi Amazon OpenSearch Tanpa Server. Pipeline adalah sumber daya yang disediakan dan dikelola oleh OpenSearch Ingestion. Anda dapat menggunakan pipeline untuk memfilter, memperkaya, mengubah, menormalkan, dan mengumpulkan data untuk analitik dan visualisasi hilir di Layanan. OpenSearch

Untuk tutorial yang menunjukkan cara menyerap data ke dalam domain OpenSearch Layanan yang disediakan, lihat. the section called "Tutorial: Menelan data ke dalam domain"

Anda akan menyelesaikan langkah-langkah berikut dalam tutorial ini:

1. Buat peran pipeline.
2. Buat koleksi.

3. Buat pipa.
4. Menelan beberapa data sampel.

Dalam tutorial, Anda akan membuat sumber daya berikut:

- Sebuah pipa bernama `ingestion-pipeline-serverless`
- Sebuah koleksi bernama `ingestion-collection` bahwa pipeline akan menulis
- Peran IAM bernama `PipelineRole` bahwa pipeline akan diasumsikan untuk menulis ke koleksi

Izin yang diperlukan

Untuk menyelesaikan tutorial ini, Anda harus memiliki izin IAM yang benar. Pengguna atau peran Anda harus memiliki kebijakan berbasis identitas terlampir dengan izin minimum berikut. Izin ini memungkinkan Anda membuat peran pipeline (`iam:Create*`), membuat atau memodifikasi collection (`aoss:*`), dan bekerja dengan pipelines (`osis:*`).

Selain itu, `iam:PassRole` izin diperlukan pada sumber daya peran pipa. Izin ini memungkinkan Anda untuk meneruskan peran pipeline ke OpenSearch Ingestion sehingga dapat menulis data ke koleksi.

```
{
   "Version":"2012-10-17",
   "Statement":[
      {
         "Effect":"Allow",
         "Resource":"*",
         "Action":[
            "osis:*",
            "iam:Create*",
            "aoss:*"
         ]
      },
      {
         "Resource":[
            "arn:aws:iam::{your-account-id}:role/PipelineRole"
         ],
         "Effect":"Allow",
         "Action":[
            "iam:PassRole"
         ]
```

```
      }
   ]
}
```

Langkah 1: Buat peran pipeline

Pertama, buat peran yang akan diasumsikan oleh pipeline untuk mengakses sink koleksi OpenSearch Tanpa Server. Anda akan menyertakan peran ini dalam konfigurasi pipeline nanti dalam tutorial ini.

Untuk membuat peran pipeline

1. Buka AWS Identity and Access Management konsol di https://console.aws.amazon.com/iamv2/.
2. Pilih Kebijakan, lalu pilih Buat kebijakan.
3. Pilih JSON dan tempelkan kebijakan berikut ke editor.

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Action": [
        "aoss:BatchGetCollection",
        "aoss:APIAccessAll"
      ],
      "Effect": "Allow",
      "Resource": "arn:aws:aoss:{region}:{your-account-id}:collection/{collection-
id}"
    },
    {
      "Action": [
        "aoss:CreateSecurityPolicy",
        "aoss:GetSecurityPolicy",
        "aoss:UpdateSecurityPolicy"
      ],
      "Effect": "Allow",
      "Resource": "*",
      "Condition": {
        "StringEquals": {
          "aoss:collection": "{collection-name}"
        }
      }
    }
```

```
  ]
}
```

4. Pilih Berikutnya, pilih Berikutnya, dan beri nama kebijakan Anda collection-pipeline-policy.
5. Pilih Buat kebijakan.
6. Selanjutnya, buat peran dan lampirkan kebijakan padanya. Pilih Peran, lalu pilih Buat peran.
7. Pilih Kebijakan kepercayaan khusus dan tempelkan kebijakan berikut ke editor:

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

8. Pilih Berikutnya. Kemudian cari dan pilih collection-pipeline-policy(yang baru saja Anda buat).
9. Pilih Berikutnya dan beri nama peran PipelineRole.
10. Pilih Buat peran.

Ingat Nama Sumber Daya Amazon (ARN) dari peran tersebut (misalnya,`arn:aws:iam::{your-account-id}:role/PipelineRole`). Anda akan membutuhkannya saat membuat pipeline Anda.

Langkah 2: Buat koleksi

Selanjutnya, buat koleksi untuk menyerap data ke dalam. Kami akan beri nama koleksinya`aingestion-collection`.

1. Arahkan ke konsol OpenSearch Layanan Amazon di https://console.aws.amazon.com/aos/home.
2. Pilih Koleksi dari navigasi kiri dan pilih Buat koleksi.
3. Sebutkan koleksi koleksi ingestion-.
4. Di bawah Pengaturan akses jaringan, ubah jenis akses ke Publik.
5. Simpan semua pengaturan lain sebagai defaultnya dan pilih Berikutnya.

6. Untuk metode Definisi, pilih JSON dan tempel kebijakan berikut ke editor. Kebijakan ini melakukan dua hal:

   - Memungkinkan peran pipeline untuk menulis ke koleksi.
   - Memungkinkan Anda membaca dari koleksi. Kemudian, setelah Anda memasukkan beberapa data sampel ke dalam pipeline, Anda akan menanyakan koleksi untuk memastikan bahwa data berhasil dicerna dan ditulis ke indeks.

```
[
  {
    "Rules": [
      {
        "Resource": [
          "index/ingestion-collection/*"
        ],
        "Permission": [
          "aoss:CreateIndex",
          "aoss:UpdateIndex",
          "aoss:DescribeIndex",
          "aoss:ReadDocument",
          "aoss:WriteDocument"
        ],
        "ResourceType": "index"
      }
    ],
    "Principal": [
      "arn:aws:iam::{your-account-id}:role/PipelineRole",
      "arn:aws:iam::{your-account-id}:role/Admin"
    ],
    "Description": "Rule 1"
  }
]
```

7. Ganti `Principal` elemen. Prinsipal pertama harus menentukan peran pipeline yang Anda buat. Yang kedua harus menentukan pengguna atau peran yang dapat Anda gunakan untuk menanyakan koleksi nanti.
8. Pilih Berikutnya. Beri nama kebijakan akses pipeline-domain-accessdan pilih Berikutnya lagi.
9. Tinjau konfigurasi koleksi Anda dan pilih Kirim.

Saat koleksi aktif, perhatikan OpenSearch titik akhir di bawah Endpoint (misalnya,`https://{collection-id}.us-east-1.aoss.amazonaws.com`). Anda akan membutuhkannya saat membuat pipeline Anda.

Langkah 3: Buat pipeline

Sekarang setelah Anda memiliki koleksi dan peran dengan hak akses yang sesuai, Anda dapat membuat pipeline.

Untuk membuat pipa

1. Di dalam konsol OpenSearch Layanan Amazon, pilih Pipelines dari panel navigasi kiri.
2. Pilih Buat pipeline.
3. Beri nama pipeline serverless-ingestion dan pertahankan pengaturan kapasitas sebagai defaultnya.
4. Dalam tutorial ini, kita akan membuat sub-pipeline sederhana `log-pipeline` yang disebut yang menggunakan plugin sumber HTTP. Plugin menerima data log dalam format array JSON. Kami akan menentukan koleksi OpenSearch Tanpa Server tunggal sebagai wastafel, dan menelan semua data ke dalam indeks. `my_logs`

   Di bawah konfigurasi Pipeline, tempelkan konfigurasi YAMAL berikut ke editor:

```
version: "2"
log-pipeline:
  source:
    http:
      path: "/${pipelineName}/test_ingestion_path"
  processor:
    - date:
        from_time_received: true
        destination: "@timestamp"
  sink:
    - opensearch:
        hosts: [ "https://{collection-id}.us-east-1.aoss.amazonaws.com" ]
        index: "my_logs"
        aws:
          sts_role_arn: "arn:aws:iam::{your-account-id}:role/PipelineRole"
          region: "us-east-1"
          serverless: true
```

5. Ganti `hosts` URL dengan titik akhir koleksi yang Anda buat di bagian sebelumnya. Ganti `sts_role_arn` parameter dengan ARN dari. `PipelineRole` Secara opsional, modifikasi. `region`
6. Pilih Validasi pipeline dan pastikan validasi berhasil.
7. Untuk kesederhanaan dalam tutorial ini, kita akan mengkonfigurasi akses publik untuk pipeline. Di bawah Jaringan, pilih Akses publik.

   Untuk informasi tentang mengonfigurasi akses VPC, lihat. the section called “Mengamankan jaringan pipa dalam VPC”
8. Tetap aktifkan penerbitan log jika Anda mengalami masalah saat menyelesaikan tutorial ini. Untuk informasi selengkapnya, lihat the section called “Memantau log”.

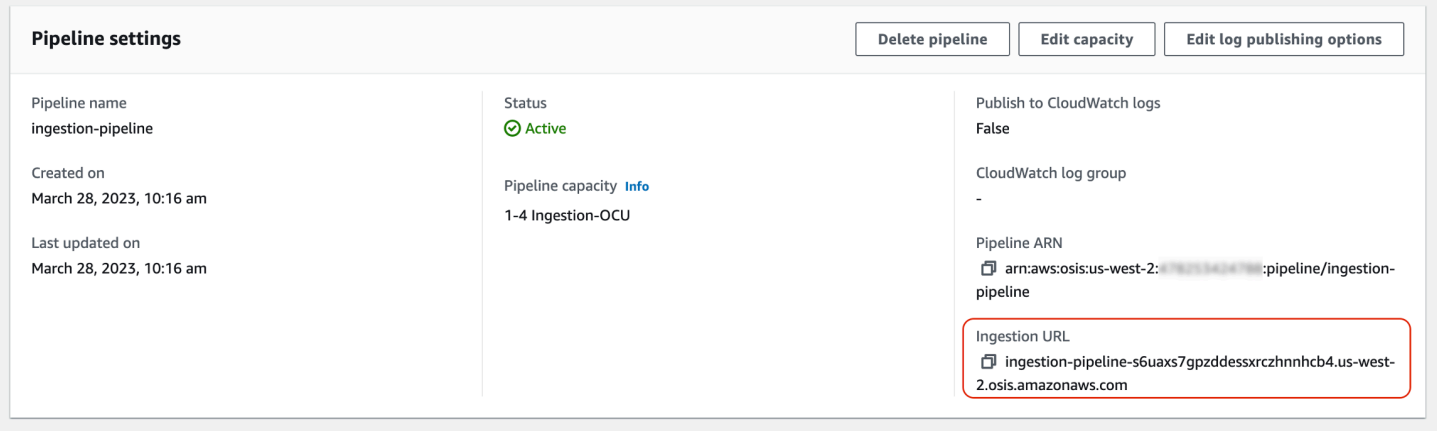
   Tentukan nama grup log berikut: `/aws/vendedlogs/OpenSearchIngestion/serverless-ingestion/audit-logs`
9. Pilih Berikutnya. Tinjau konfigurasi pipeline Anda dan pilih Create pipeline. Pipa membutuhkan waktu 5-10 menit untuk menjadi aktif.

Langkah 4: Menelan beberapa data sampel

Ketika status pipeline`Active`, Anda dapat mulai menelan data ke dalamnya. Anda harus menandatangani semua permintaan HTTP ke pipeline menggunakan Signature Version 4. Gunakan alat HTTP seperti Postman atau awscurl untuk mengirim beberapa data ke pipeline. Seperti halnya pengindeksan data langsung ke koleksi, menelan data ke dalam pipeline selalu memerlukan peran IAM atau kunci akses IAM dan kunci rahasia.

> **Note**
>
> Kepala sekolah yang menandatangani permintaan harus memiliki izin `osis:Ingest` IAM.

Pertama, dapatkan URL konsumsi dari halaman pengaturan Pipeline:



Kemudian, konsumsi beberapa data sampel. Permintaan contoh berikut menggunakan awscurl untuk mengirim satu file log ke indeks: `my_logs`

```
awscurl --service osis --region us-east-1 \
    -X POST \
    -H "Content-Type: application/json" \
    -d
 '[{"time":"2014-08-11T11:40:13+00:00","remote_addr":"122.226.223.69","status":"404","request":
 http://www.k2proxy.com//hello.html HTTP/1.1","http_user_agent":"Mozilla/4.0
 (compatible; WOW64; SLCC2;)"}]' \
    https://{pipeline-endpoint}.us-east-1.osis.amazonaws.com/log-pipeline/
test_ingestion_path
```

Anda harus melihat `200 OK` tanggapan.

Sekarang, kueri `my_logs` indeks untuk memastikan bahwa entri log berhasil dicerna:

```
awscurl --service aoss --region us-east-1 \
     -X GET \
     https://{collection-id}.us-east-1.aoss.amazonaws.com/my_logs/_search | json_pp
```

Sampel respon:

```
{
   "took":348,
   "timed_out":false,
   "_shards":{
      "total":0,
      "successful":0,
```

```
      "skipped":0,
      "failed":0
   },
   "hits":{
      "total":{
         "value":1,
         "relation":"eq"
      },
      "max_score":1.0,
      "hits":[
         {
            "_index":"my_logs",
            "_id":"1%3A0%3ARJgDvIcBTy5m12xrKE-y",
            "_score":1.0,
            "_source":{
               "time":"2014-08-11T11:40:13+00:00",
               "remote_addr":"122.226.223.69",
               "status":"404",
               "request":"GET http://www.k2proxy.com//hello.html HTTP/1.1",
               "http_user_agent":"Mozilla/4.0 (compatible; WOW64; SLCC2;)",
               "@timestamp":"2023-04-26T05:22:16.204Z"
            }
         }
      ]
   }
}
```


Sumber daya terkait

Tutorial ini menyajikan kasus penggunaan sederhana menelan satu dokumen melalui HTTP. Dalam skenario produksi, Anda akan mengonfigurasi aplikasi klien Anda (seperti Fluent Bit, Kubernetes, atau OpenTelemetry Collector) untuk mengirim data ke satu atau beberapa pipeline. Saluran pipa Anda kemungkinan akan lebih kompleks daripada contoh sederhana dalam tutorial ini.

Untuk mulai mengonfigurasi klien Anda dan menelan data, lihat sumber daya berikut:

- Membuat dan mengelola jaringan pipa
- Mengkonfigurasi klien Anda untuk mengirim data ke Ingestion OpenSearch
- Dokumentasi Data Prepper

Ikhtisar fitur pipeline di Amazon OpenSearch Ingestion

Amazon OpenSearch Ingestion menyediakan saluran pipa, yang terdiri dari sumber, buffer, nol atau lebih prosesor, dan satu atau lebih sink. Saluran pipa konsumsi ditenagai oleh Data Prepper sebagai mesin data. Untuk ikhtisar berbagai komponen pipa, lihatthe section called “Konsep utama”.

Bagian berikut memberikan gambaran umum tentang beberapa fitur yang paling umum digunakan di Amazon OpenSearch Ingestion.

> **Note**
>
> Ini bukan daftar lengkap fitur yang tersedia untuk saluran pipa. Untuk dokumentasi komprehensif dari semua fungsi pipeline yang tersedia, lihat dokumentasi Data Prepper. Perhatikan bahwa OpenSearch Ingestion menempatkan beberapa kendala pada plugin dan opsi yang dapat Anda gunakan. Untuk informasi selengkapnya, lihat the section called “Plugin dan opsi yang didukung”.

Topik

Buffering persisten

Buffer persisten menyimpan data Anda dalam buffer berbasis disk di beberapa Availability Zone untuk menambah daya tahan pada data Anda. Anda dapat menggunakan buffering persisten untuk menyerap data untuk semua sumber berbasis push yang didukung tanpa perlu menyiapkan buffer mandiri. Ini termasuk HTTP dan OpenTelemetry sumber untuk log, jejak, dan metrik.

Untuk mengaktifkan buffering persisten, pilih Aktifkan buffer persisten saat membuat atau memperbarui pipeline. Untuk informasi lebih lanjut, lihatthe section called “Membuat jaringan pipa”.

OpenSearch Ingestion secara otomatis menentukan kapasitas buffering yang diperlukan berdasarkan Unit OpenSearch Komputasi Tertelan (OCU Ingestion) yang Anda tentukan untuk pipeline.

Secara default, pipeline menggunakan file Kunci milik AWS untuk mengenkripsi data buffer. Pipeline ini tidak memerlukan izin tambahan untuk peran pipeline. Sebagai alternatif, Anda dapat menentukan kunci terkelola pelanggan dan menambahkan izin IAM berikut ke peran pipeline:

```
{
    "Version": "2012-10-17",
    "Statement": [
        {
            "Sid": "KeyAccess",
            "Effect": "Allow",
            "Action": [
              "kms:Decrypt",
              "kms:GenerateDataKeyWithoutPlaintext"
            ],
            "Resource": "arn:aws:kms:{region}:{aws-account-
id}:key/1234abcd-12ab-34cd-56ef-1234567890ab"
        }
    ]
}
```

Untuk informasi selengkapnya, lihat Kunci terkelola pelanggan di Panduan AWS Key Management Service Pengembang.

> **Note**
>
> Jika Anda menonaktifkan buffering persisten, pipeline Anda akan diperbarui untuk berjalan sepenuhnya pada buffering dalam memori.

Menyetel ukuran payload permintaan maksimum

Jika Anda mengaktifkan buffering persisten untuk pipeline, Anda memiliki opsi untuk menyetel ukuran payload permintaan maksimum. Pengaturan ini membatasi ukuran catatan yang dikirim ke wastafel dalam satu permintaan, sehingga menghindari pengiriman permintaan besar. Untuk menyetel ukuran muatan maksimum, atur `max_request_length` opsi dalam konfigurasi sumber. Sama seperti buffering persisten, opsi ini hanya didukung untuk HTTP dan OpenTelemetry sumber untuk log, jejak, dan metrik.

Satu-satunya nilai yang valid untuk `max_request_length` opsi ini adalah 1mb, 1.5mb, 2mb, 2.5mb, 3mb, 3.5mb, dan 4mb. Jika Anda menentukan nilai yang berbeda, Anda menerima kesalahan.

Contoh berikut menunjukkan cara mengkonfigurasi ukuran payload maksimum dalam konfigurasi pipeline:

```
...
log-pipeline:
  source:
    http:
      path: "/${pipelineName}/logs"
      max_request_length: "4mb"
  processor:
...
```

Jika Anda mengaktifkan buffering persisten dan tidak menentukan `max_request_length` opsi, nilai defaultnya menjadi 1mb.

Memisahkan

Anda dapat mengonfigurasi pipeline OpenSearch Ingestion untuk membagi peristiwa masuk menjadi sub-pipeline, memungkinkan Anda melakukan berbagai jenis pemrosesan pada acara masuk yang sama.

Contoh pipeline berikut membagi peristiwa yang masuk menjadi dua sub-pipeline. Setiap sub-pipeline menggunakan prosesornya sendiri untuk memperkaya dan memanipulasi data, dan kemudian mengirimkan data ke indeks yang berbeda. OpenSearch

```
version: "2"
log-pipeline:
  source:
    http:
    ...
  sink:
    - pipeline:
        name: "logs_enriched_one_pipeline"
    - pipeline:
        name: "logs_enriched_two_pipeline"

logs_enriched_one_pipeline:
  source:
    log-pipeline
```

```
  processor:
   ...
  sink:
    - opensearch:
        # Provide a domain or collection endpoint
        # Enable the 'serverless' flag if the sink is an OpenSearch Serverless
 collection
        aws:
          ...
        index: "enriched_one_logs"

logs_enriched_two_pipeline:
  source:
    log-pipeline
  processor:
   ...
  sink:
    - opensearch:
        # Provide a domain or collection endpoint
        # Enable the 'serverless' flag if the sink is an OpenSearch Serverless
 collection
        aws:
          ...
          index: "enriched_two_logs"
```

Mengikat

Anda dapat menghubungkan beberapa sub-pipeline bersama-sama untuk melakukan pemrosesan dan pengayaan data dalam potongan. Dengan kata lain, Anda dapat memperkaya acara yang masuk dengan kemampuan pemrosesan tertentu dalam satu sub-pipeline, kemudian mengirimkannya ke sub-pipeline lain untuk pengayaan tambahan dengan prosesor yang berbeda, dan akhirnya mengirimkannya ke wastafelnya. OpenSearch

Dalam contoh berikut, `log_pipeline` sub-pipeline memperkaya peristiwa log masuk dengan satu set prosesor, kemudian mengirimkan acara ke indeks bernama. OpenSearch `enriched_logs` Pipeline mengirimkan peristiwa yang sama ke `log_advanced_pipeline` sub-pipeline, yang memprosesnya dan mengirimkannya ke OpenSearch indeks yang berbeda bernama`enriched_advanced_logs`.

```
version: "2"
log-pipeline:
  source:
```

```
    http:
    ...
  processor:
    ...
  sink:
    - opensearch:
        # Provide a domain or collection endpoint
        # Enable the 'serverless' flag if the sink is an OpenSearch Serverless
 collection
        aws:
          ...
          index: "enriched_logs"
    - pipeline:
        name: "log_advanced_pipeline"

log_advanced_pipeline:
  source:
    log-pipeline
  processor:
   ...
  sink:
    - opensearch:
        # Provide a domain or collection endpoint
        # Enable the 'serverless' flag if the sink is an OpenSearch Serverless
 collection
        aws:
          ...
          index: "enriched_advanced_logs"
```

Antrean surat mati

Antrian surat mati (DLQ) adalah tujuan untuk peristiwa yang gagal ditulis oleh pipa ke wastafel. Di OpenSearch Ingestion, Anda harus menentukan bucket Amazon S3 dengan izin tulis yang sesuai untuk digunakan sebagai DLQ. Anda dapat menambahkan konfigurasi DLQ ke setiap wastafel di dalam pipa. Saat pipeline menemukan kesalahan penulisan, ia membuat objek DLQ di bucket S3 yang dikonfigurasi. Objek DLQ ada dalam file JSON sebagai array peristiwa gagal.

Pipeline menulis peristiwa ke DLQ ketika salah satu dari kondisi berikut terpenuhi:

- `max_retries`Untuk OpenSearch wastafel sudah habis. OpenSearch Tertelan membutuhkan minimal 16 untuk opsi ini.
- Peristiwa ditolak oleh wastafel karena kondisi kesalahan.

Konfigurasi

Untuk mengonfigurasi antrian huruf mati untuk sub-pipeline, tentukan `dlq` opsi dalam konfigurasi wastafel: `opensearch`

```
apache-log-pipeline:
  ...
  sink:
    opensearch:
      dlq:
        s3:
          bucket: "my-dlq-bucket"
          key_path_prefix: "dlq-files"
          region: "us-west-2"
          sts_role_arn: "arn:aws:iam::123456789012:role/dlq-role"
```

File yang ditulis ke DLQ S3 ini akan memiliki pola penamaan berikut:

```
dlq-v${version}-${pipelineName}-${pluginId}-${timestampIso8601}-${uniqueId}
```

Untuk informasi selengkapnya, lihat Dead-Letter Queues (DLQ).

Untuk instruksi untuk mengonfigurasi `sts_role_arn` peran, lihatthe section called “Menulis ke antrian surat mati”.

Contoh

Perhatikan contoh file DLQ berikut:

```
dlq-v2-apache-log-pipeline-opensearch-2023-04-05T15:26:19.152938Z-e7eb675a-
f558-4048-8566-dac15a4f8343
```

Berikut adalah contoh data yang gagal ditulis ke wastafel, dan dikirim ke bucket DLQ S3 untuk analisis lebih lanjut:

```
Record_0
pluginId            "opensearch"
pluginName          "opensearch"
pipelineName        "apache-log-pipeline"
failedData
```

```
index    "logs"
indexId   null
status    0
message  "Number of retries reached the limit of max retries (configured value 15)"
document
log      "sample log"
timestamp     "2023-04-14T10:36:01.070Z"

Record_1
pluginId            "opensearch"
pluginName          "opensearch"
pipelineName        "apache-log-pipeline"
failedData
index               "logs"
indexId   null
status    0
message  "Number of retries reached the limit of max retries (configured value 15)"
document
log                 "another sample log"
timestamp           "2023-04-14T10:36:01.071Z"
```

Manajemen indeks

Amazon OpenSearch Ingestion memiliki banyak kemampuan manajemen indeks, termasuk yang berikut ini.

Membuat indeks

Anda dapat menentukan nama indeks di wastafel pipa dan OpenSearch Ingestion membuat indeks saat menyediakan pipeline. Jika indeks sudah ada, pipeline menggunakannya untuk mengindeks peristiwa yang masuk. Jika Anda menghentikan dan memulai ulang pipeline, atau jika Anda memperbarui konfigurasi YAML-nya, pipeline akan mencoba membuat indeks baru jika belum ada. Pipeline tidak akan pernah bisa menghapus indeks.

Contoh berikut sink membuat dua indeks saat pipeline disediakan:

```
sink:
  - opensearch:
      index: apache_logs
  - opensearch:
      index: nginx_logs
```

Menghasilkan nama dan pola indeks

Anda dapat menghasilkan nama indeks dinamis dengan menggunakan variabel dari bidang peristiwa yang masuk. Dalam konfigurasi sink, gunakan format `string${}` untuk memberi sinyal interpolasi string, dan gunakan penunjuk JSON untuk mengekstrak bidang dari peristiwa. Pilihan untuk `index_type` adalah `custom` ataumanagement_disabled. Karena `index_type` default untuk OpenSearch domain dan `custom` `management_disabled` untuk koleksi OpenSearch Tanpa Server, itu dapat dibiarkan tidak disetel.

Misalnya, pipeline berikut memilih `metadataType` bidang dari peristiwa yang masuk untuk menghasilkan nama indeks.

```
pipeline:
  ...
  sink:
    opensearch:
      index: "metadata-${metadataType}"
```

Konfigurasi berikut terus menghasilkan indeks baru setiap hari atau setiap jam.

```
pipeline:
  ...
  sink:
    opensearch:
      index: "metadata-${metadataType}-%{yyyy.MM.dd}"

pipeline:
  ...
  sink:
    opensearch:
      index: "metadata-${metadataType}-%{yyyy.MM.dd.HH}"
```

Nama indeks juga bisa berupa string biasa dengan pola tanggal-waktu sebagai akhiran, seperti. `my-index-%{yyyy.MM.dd}` Ketika sink mengirim data ke OpenSearch, itu menggantikan pola tanggal-waktu dengan waktu UTC dan membuat indeks baru untuk setiap hari, seperti. `my-index-2022.01.25` Untuk informasi lebih lanjut, lihat DateTimeFormatterkelas.

Nama indeks ini juga dapat berupa string yang diformat (dengan atau tanpa akhiran pola tanggal-waktu), seperti. `my-${index}-name` Ketika wastafel mengirim data ke OpenSearch, itu menggantikan `"${index}"` bagian dengan nilai dalam acara yang sedang diproses. Jika

formatnya`"${index1/index2/index3}"`, itu menggantikan bidang `index1/index2/index3` dengan nilainya dalam acara tersebut.

Menghasilkan ID dokumen

Pipeline dapat menghasilkan ID dokumen saat mengindeks dokumen ke OpenSearch. Ini dapat menyimpulkan ID dokumen ini dari bidang dalam peristiwa yang masuk.

Contoh ini menggunakan `uuid` bidang dari peristiwa yang masuk untuk menghasilkan ID dokumen.

```
pipeline:
  ...
  sink:
    opensearch:
      index_type: custom
      index: "metadata-${metadataType}-%{yyyy.MM.dd}"
      document_id_field: "uuid"
```

Dalam contoh berikut, prosesor Tambah entri menggabungkan bidang `uuid` dan `other_field` dari peristiwa yang masuk untuk menghasilkan ID dokumen.

`create`Tindakan ini memastikan bahwa dokumen dengan ID identik tidak ditimpa. Pipeline menjatuhkan dokumen duplikat tanpa percobaan ulang atau acara DLQ. Ini adalah harapan yang masuk akal bagi penulis pipeline yang menggunakan tindakan ini, karena tujuannya adalah untuk menghindari memperbarui dokumen yang ada.

```
pipeline:
  ...
  processor:
   - add_entries:
      entries:
        - key: "my_doc_id_field"
          format: "${uuid}-${other_field}"
  sink:
    - opensearch:
       ...
       action: "create"
       document_id_field: "my_doc_id_field"
```

Anda mungkin ingin menyetel ID dokumen acara ke bidang dari sub-objek. Dalam contoh berikut, plugin OpenSearch sink menggunakan sub-objek `info/id` untuk menghasilkan ID dokumen.

```
sink:
  - opensearch:
       ...
       document_id_field: info/id
```

Mengingat peristiwa berikut, pipeline akan menghasilkan dokumen dengan `_id` bidang yang disetel `kejson001`:

```
{
   "fieldA":"arbitrary value",
   "info":{
      "id":"json001",
      "fieldA":"xyz",
      "fieldB":"def"
   }
}
```

Menghasilkan ID perutean

Anda dapat menggunakan `routing_field` opsi dalam plugin OpenSearch sink untuk mengatur nilai properti routing dokumen (`_routing`) ke nilai dari peristiwa yang masuk.

Routing mendukung sintaks pointer JSON, sehingga bidang bersarang juga tersedia, bukan hanya bidang tingkat atas.

```
sink:
  - opensearch:
       ...
       routing_field: metadata/id
       document_id_field: id
```

Mengingat peristiwa berikut, plugin menghasilkan dokumen dengan `_routing` bidang diatur `keabcd`:

```
{
   "id":"123",
   "metadata":{
      "id":"abcd",
      "fieldA":"valueA"
   },
   "fieldB":"valueB"
}
```

Untuk petunjuk membuat templat indeks yang dapat digunakan pipeline selama pembuatan indeks, lihat Templat indeks.

E nd-to-end pengakuan

OpenSearch Penyerapan memastikan daya tahan dan keandalan data dengan melacak pengirimannya dari sumber ke bak cuci di jaringan pipa tanpa kewarganegaraan menggunakan pengakuan. end-to-end Saat ini, hanya plugin sumber S3 yang mendukung end-to-end pengakuan.

Dengan end-to-end pengakuan, plugin sumber pipeline membuat set pengakuan untuk memantau sekumpulan peristiwa. Ini menerima pengakuan positif ketika peristiwa itu berhasil dikirim ke wastafel mereka, atau pengakuan negatif ketika salah satu peristiwa tidak dapat dikirim ke wastafel mereka.

Jika terjadi kegagalan atau kerusakan komponen pipa, atau jika sumber gagal menerima pengakuan, sumber akan habis waktu dan mengambil tindakan yang diperlukan seperti mencoba kembali atau mencatat kegagalan. Jika pipeline memiliki beberapa sink atau beberapa sub-pipeline yang dikonfigurasi, pengakuan tingkat peristiwa dikirim hanya setelah acara dikirim ke semua sink di semua sub-pipeline. Jika wastafel memiliki DLQ yang dikonfigurasi, end-to-end pengakuan juga melacak peristiwa yang ditulis ke DLQ.

Untuk mengaktifkan end-to-end pengakuan, sertakan `acknowledgments` opsi dalam konfigurasi sumber:

```
s3-pipeline:
  source:
    s3:
      acknowledgments: true
...
```

Sumber tekanan balik

Pipa dapat mengalami tekanan balik saat sibuk memproses data, atau jika sink sementara turun atau lambat untuk menelan data. OpenSearch Ingestion memiliki cara yang berbeda untuk menangani tekanan balik tergantung pada plugin sumber yang digunakan pipa.

Sumber HTTP

Saluran pipa yang menggunakan plugin sumber HTTP menangani tekanan balik secara berbeda tergantung pada komponen pipa mana yang padat:

- Buffer — Ketika buffer penuh, pipeline mulai mengembalikan status HTTP `REQUEST_TIMEOUT` dengan kode kesalahan 408 kembali ke titik akhir sumber. Saat buffer dibebaskan, pipeline mulai memproses peristiwa HTTP lagi.
- Thread sumber — Ketika semua thread sumber HTTP sibuk mengeksekusi permintaan dan ukuran antrian permintaan yang belum diproses telah melebihi jumlah maksimum permintaan yang diizinkan, pipeline mulai mengembalikan status HTTP `TOO_MANY_REQUESTS` dengan kode kesalahan 429 kembali ke titik akhir sumber. Ketika antrian permintaan turun di bawah ukuran antrian maksimum yang diizinkan, pipeline mulai memproses permintaan lagi.

Sumber oTel

Ketika buffer penuh untuk pipeline yang menggunakan OpenTelemetry sumber (log OTel, metrik OTel, dan jejak OTel), pipeline mulai mengembalikan status HTTP `REQUEST_TIMEOUT` dengan kode kesalahan 408 ke titik akhir sumber. Saat buffer dibebaskan, pipeline mulai memproses peristiwa lagi.

Sumber S3

Ketika buffer penuh untuk saluran pipa dengan sumber S3, saluran pipa berhenti memproses pemberitahuan SQS. Saat buffer dibebaskan, saluran pipa mulai memproses notifikasi lagi.

Jika sink mati atau tidak dapat menyerap data dan end-to-end pengakuan diaktifkan untuk sumber, pipeline berhenti memproses notifikasi SQS hingga menerima pengakuan yang berhasil dari semua sink.

Membuat pipa Amazon OpenSearch Ingestion

Pipeline adalah mekanisme yang digunakan Amazon OpenSearch Ingestion untuk memindahkan data dari sumbernya (dari mana data berasal) ke wastafelnya (ke mana data pergi). Dalam OpenSearch Ingestion, wastafel akan selalu menjadi domain OpenSearch Layanan Amazon tunggal, sedangkan sumber data Anda bisa berupa klien seperti Amazon S3, Fluent Bit, atau Collector. OpenTelemetry

Untuk informasi selengkapnya, lihat Pipelines dalam OpenSearch dokumentasi.

Topik

- Prasyarat dan peran yang diperlukan
- Izin diperlukan

- Menentukan versi pipeline
- Menentukan jalur konsumsi
- Membuat jaringan pipa
- Melacak status pembuatan pipa
- Menggunakan cetak biru untuk membuat pipeline

Prasyarat dan peran yang diperlukan

Untuk membuat pipa OpenSearch Ingestion, Anda harus memiliki sumber daya berikut:

- Peran IAM yang akan diasumsikan oleh OpenSearch Ingestion untuk menulis ke wastafel. Anda akan menyertakan peran ARN ini dalam konfigurasi pipeline Anda.
- Domain OpenSearch Layanan atau koleksi OpenSearch Tanpa Server untuk bertindak sebagai wastafel. Jika Anda menulis ke domain, itu harus menjalankan OpenSearch 1.0 atau yang lebih baru, atau Elasticsearch 7.4 atau yang lebih baru. Wastafel harus memiliki kebijakan akses yang memberikan izin yang sesuai untuk peran pipeline IAM Anda.

Untuk petunjuk untuk membuat sumber daya ini, lihat topik berikut:

- the section called “Memberikan akses jaringan pipa ke domain”
- the section called “Memberikan akses jaringan pipa ke koleksi”

> **Note**
>
> Jika Anda menulis ke domain yang menggunakan kontrol akses berbutir halus, ada langkah-langkah tambahan yang perlu Anda selesaikan. Lihat the section called “Langkah 3: Petakan peran pipeline (hanya untuk domain yang menggunakan kontrol akses berbutir halus)”.

Izin diperlukan

OpenSearch Ingestion menggunakan izin IAM berikut untuk membuat pipeline:

- `osis:CreatePipeline`— Buat pipa.
- `osis:ValidatePipeline`— Periksa apakah konfigurasi pipeline valid.

- `iam:PassRole`— Lewati peran pipeline ke OpenSearch Ingestion sehingga dapat menulis data ke domain. Izin ini harus ada pada sumber daya peran pipeline (ARN yang Anda tentukan untuk `sts_role_arn` opsi dalam konfigurasi pipeline), atau hanya * jika Anda berencana untuk menggunakan peran yang berbeda di setiap pipeline.

Misalnya, kebijakan berikut memberikan izin untuk membuat pipeline:

```
{
   "Version":"2012-10-17",
   "Statement":[
      {
         "Effect":"Allow",
         "Resource":"*",
         "Action":[
            "osis:CreatePipeline",
            "osis:ListPipelineBlueprints",
            "osis:ValidatePipeline"
         ]
      },
      {
         "Resource":[
            "arn:aws:iam::{your-account-id}:role/{pipeline-role}"
         ],
         "Effect":"Allow",
         "Action":[
            "iam:PassRole"
         ]
      }
   ]
}
```

OpenSearch Ingestion juga menyertakan izin yang dipanggil`osis:Ingest`, yang diperlukan untuk mengirim permintaan yang ditandatangani ke pipeline menggunakan Signature Version 4. Untuk informasi selengkapnya, lihat the section called “Membuat peran konsumsi”.

> **Note**
>
> Selain itu, pengguna pertama yang membuat pipeline di akun harus memiliki izin untuk `iam:CreateServiceLinkedRole` tindakan tersebut. Untuk informasi selengkapnya, lihat sumber daya peran pipeline.

Untuk informasi selengkapnya tentang setiap izin, lihat Tindakan, sumber daya, dan kunci kondisi untuk OpenSearch Tertelan di Referensi Otorisasi Layanan.

Menentukan versi pipeline

Saat Anda mengonfigurasi pipeline, Anda harus menentukan versi utama Data Prepper yang akan dijalankan pipeline. Untuk menentukan versi, sertakan `version` opsi dalam konfigurasi pipeline Anda:

```
version: "2"
log-pipeline:
  source:
    ...
```

Saat Anda memilih Buat, OpenSearch Ingestion menentukan versi minor terbaru yang tersedia dari versi utama yang Anda tentukan, dan menyediakan pipeline dengan versi tersebut. Misalnya, jika Anda menentukan`version: "2"`, dan versi terbaru yang didukung dari Data Prepper adalah 2.1.1, OpenSearch Ingestion menyediakan pipeline Anda dengan versi 2.1.1. Kami tidak menampilkan versi minor yang sedang dijalankan pipeline Anda secara publik.

Untuk meningkatkan pipeline Anda saat versi utama baru dari Data Prepper tersedia, edit konfigurasi pipeline dan tentukan versi baru. Anda tidak dapat menurunkan versi pipeline ke versi sebelumnya.

> **Note**
>
> OpenSearch Ingestion tidak segera mendukung versi baru dari Data Prepper segera setelah dirilis. Akan ada beberapa jeda antara saat versi baru tersedia untuk umum dan saat didukung di OpenSearch Ingestion. Selain itu, OpenSearch Ingestion mungkin secara eksplisit tidak mendukung versi mayor atau minor tertentu sama sekali. Untuk daftar lengkap, lihatthe section called “Versi Data Prepper yang Didukung”.

Setiap kali Anda membuat perubahan pada pipeline yang memulai penerapan biru/hijau, OpenSearch Ingestion dapat memutakhirkannya ke versi minor terbaru dari versi utama yang saat ini dikonfigurasi dalam file YAMAL pipeline. Untuk informasi lebih lanjut, lihatthe section called “Penerapan biru/hijau untuk pembaruan saluran pipa”. OpenSearch Ingestion tidak dapat mengubah versi utama pipeline Anda kecuali Anda secara eksplisit memperbarui `version` opsi dalam konfigurasi pipeline.

Menentukan jalur konsumsi

Untuk sumber berbasis tarik seperti OtEL trace dan metrik OTel, OpenSearch Ingestion memerlukan opsi tambahan dalam konfigurasi sumber Anda. `path` Path adalah string seperti`/log/ingest`, yang mewakili jalur URI untuk konsumsi. Path ini mendefinisikan URI yang Anda gunakan untuk mengirim data ke pipeline.

Misalnya, Anda menentukan sub-pipeline entri berikut untuk pipeline konsumsi bernama: `logs`

```
entry-pipeline:
  source:
    http:
      path: "/my/test_path"
```

Saat Anda memasukkan data ke dalam pipeline, Anda harus menentukan titik akhir berikut dalam konfigurasi klien Anda:. `https://logs-abcdefgh.us-west-2.osis.amazonaws.com/my/test_path`

Jalur harus dimulai dengan garis miring (/) dan dapat berisi karakter khusus '-', '_', ' . ', dan'/', serta `${pipelineName}` placeholder. Jika Anda menggunakan `${pipelineName}` (sepertipath: "/${pipelineName}/test_path"), variabel diganti dengan nama sub-pipeline terkait. Dalam contoh ini, itu akan terjadi`https://logs.us-west-2.osis.amazonaws.com/entry-pipeline/test_path`.

Membuat jaringan pipa

Bagian ini menjelaskan cara membuat pipeline OpenSearch Ingestion menggunakan konsol OpenSearch Layanan dan. AWS CLI

Konsol

Untuk membuat pipa

1. Masuk ke konsol OpenSearch Layanan Amazon di https://console.aws.amazon.com/aos/home.
2. Pilih Pipelines di panel navigasi kiri dan pilih Create pipeline.
3. Masukkan nama untuk alur.
4. (Opsional) Pilih Aktifkan buffer persisten. Buffer persisten menyimpan data Anda dalam buffer berbasis disk di beberapa AZ. Untuk informasi lebih lanjut, lihat Buffering persisten. Jika Anda

mengaktifkan buffer persisten, pilih AWS Key Management Service kunci untuk mengenkripsi data buffer.

5. Konfigurasikan kapasitas pipeline minimum dan maksimum di Ingestion OpenSearch Compute Units (OCU). Untuk informasi selengkapnya, lihat the section called “Penskalaan pipa”.
6. Di bawah konfigurasi Pipeline, berikan konfigurasi pipeline Anda dalam format YAMG. File konfigurasi pipeline tunggal dapat berisi 1-10 sub-pipeline. Setiap sub-pipa adalah kombinasi dari satu sumber, nol atau lebih prosesor, dan satu wastafel. Untuk OpenSearch Ingestion, wastafel harus selalu menjadi domain OpenSearch Layanan. Untuk daftar opsi yang didukung, lihatthe section called “Plugin dan opsi yang didukung”.

> **Note**
>
> Anda harus menyertakan `sts_role_arn` dan `sigv4` opsi di setiap sub-pipeline. Pipeline mengasumsikan aturan yang ditentukan `sts_role_arn` untuk menandatangani permintaan ke domain. Untuk informasi selengkapnya, lihat the section called “Memberikan akses jaringan pipa ke domain”.

Contoh file konfigurasi berikut menggunakan sumber HTTP dan plugin Grok untuk memproses data log tidak terstruktur dan mengirimkannya ke domain Layanan. OpenSearch Sub-pipeline diberi nama`log-pipeline`.

```
version: "2"
log-pipeline:
  source:
    http:
      path: "/log/ingest"
  processor:
    - grok:
        match:
          log: [ '%{COMMONAPACHELOG}' ]
    - date:
        from_time_received: true
        destination: "@timestamp"
  sink:
    - opensearch:
        hosts: [ "https://search-my-domain.us-east-1.es.amazonaws.com" ]
        index: "apache_logs"
        aws:
```

```
        sts_role_arn: "arn:aws:iam::123456789012:role/{pipeline-role}"
        region: "us-east-1"
```

> **Note**
>
> Jika Anda menentukan beberapa sink dalam definisi pipeline YAMG, semuanya harus merupakan domain OpenSearch Layanan yang sama. Pipeline OpenSearch Ingestion tidak dapat menulis ke beberapa domain yang berbeda.

Anda dapat membuat konfigurasi pipeline sendiri, atau memilih Unggah file dan mengimpor konfigurasi yang ada untuk pipeline Persiapan Data yang dikelola sendiri. Atau, Anda dapat menggunakan cetak biru konfigurasi.

7. Setelah mengonfigurasi pipeline, pilih Validasi pipeline untuk mengonfirmasi bahwa konfigurasi sudah benar. Jika validasi gagal, perbaiki kesalahan dan jalankan kembali validasi.
8. Di bawah Jaringan, pilih akses VPC atau Akses publik. Jika Anda memilih Akses publik, lewati ke langkah berikutnya. Jika Anda memilih akses VPC, konfigurasikan pengaturan berikut:

Pengaturan	Deskripsi
VPC	Pilih ID virtual private cloud (VPC) yang ingin Anda gunakan. VPC dan pipeline harus sama. Wilayah AWS
Subnet	Pilih satu atau lebih subnet. OpenSearch Layanan akan menempatkan titik akhir VPC dan antarmuka jaringan elastis di subnet.
Grup keamanan	Pilih satu atau beberapa grup keamanan VPC yang memungkinkan aplikasi Anda mencapai pipeline OpenSearch Ingestion pada port (80 atau 443) dan protokol (HTTP atau HTTPS) yang diekspos oleh pipeline.

Untuk informasi selengkapnya, lihat the section called “Mengamankan jaringan pipa dalam VPC”.

9. (Opsional) Di bawah Tag, tambahkan satu atau beberapa tag (pasangan nilai kunci) ke pipeline Anda. Untuk informasi selengkapnya, lihat the section called “Menandai”.
10. (Opsional) Di bawah opsi penerbitan Log, aktifkan penerbitan log pipeline ke Amazon CloudWatch Logs. Kami menyarankan Anda mengaktifkan penerbitan log sehingga Anda dapat

lebih mudah memecahkan masalah pipeline. Untuk informasi selengkapnya, lihat the section called “Memantau log”.

11. Pilih Berikutnya.
12. Tinjau konfigurasi pipeline Anda dan pilih Buat.

OpenSearch Ingestion menjalankan proses asinkron untuk membangun pipeline. Setelah status pipeline`Active`, Anda dapat mulai menelan data.

AWS CLI

Perintah create-pipeline menerima konfigurasi pipeline sebagai string atau dalam file.yaml. Jika Anda memberikan konfigurasi sebagai string, setiap baris baru harus diloloskan. `\n` Misalnya, `"log-pipeline:\n source:\n http:\n processor:\n - grok:\n ...`

Perintah contoh berikut membuat pipeline dengan konfigurasi berikut:

- Minimal 4 OCU Tertelan, maksimum 10 OCU Tertelan
- Disediakan dalam virtual private cloud (VPC)
- Penerbitan log diaktifkan

```
aws osis create-pipeline \
  --pipeline-name my-pipeline \
  --min-units 4 \
  --max-units 10 \
  --log-publishing-options
 IsLoggingEnabled=true,CloudWatchLogDestination={LogGroup="MyLogGroup"} \
  --vpc-options
 SecurityGroupIds={sg-12345678,sg-9012345},SubnetIds=subnet-1212234567834asdf \
  --pipeline-configuration-body "file://pipeline-config.yaml"
```

OpenSearch Ingestion menjalankan proses asinkron untuk membangun pipeline. Setelah status pipeline`Active`, Anda dapat mulai menelan data. Untuk memeriksa status pipa, gunakan GetPipelineperintah.

OpenSearch API Tertelan

Untuk membuat pipeline OpenSearch Ingestion menggunakan API OpenSearch Ingestion, panggil operasi. CreatePipeline

Setelah pipeline berhasil dibuat, Anda dapat mengonfigurasi klien dan mulai memasukkan data ke dalam domain OpenSearch Layanan. Untuk informasi selengkapnya, lihat the section called “Bekerja dengan integrasi pipa”.

Melacak status pembuatan pipa

Anda dapat melacak status pipa saat OpenSearch Ingestion menyediakannya dan menyiapkannya untuk menyerap data.

Konsol

Setelah Anda awalnya membuat pipeline, ia melewati beberapa tahap saat OpenSearch Ingestion mempersiapkannya untuk menelan data. Untuk melihat berbagai tahapan pembuatan pipeline, pilih nama pipeline untuk melihat halaman pengaturan Pipeline. Di bawah Status, pilih Lihat detail.

Pipeline melewati tahapan berikut sebelum tersedia untuk menelan data:

- Validasi - Memvalidasi konfigurasi pipeline. Ketika tahap ini selesai, semua validasi telah berhasil.
- Menciptakan lingkungan — Mempersiapkan dan menyediakan sumber daya. Ketika tahap ini selesai, lingkungan pipa baru telah dibuat.
- Menyebarkan pipa - Menyebarkan pipa. Ketika tahap ini selesai, pipa telah berhasil digunakan.
- Periksa kesehatan pipa — Memeriksa kesehatan pipa. Ketika tahap ini selesai, semua pemeriksaan kesehatan telah berlalu.
- Aktifkan lalu lintas - Mengaktifkan pipeline untuk menyerap data. Ketika tahap ini selesai, Anda dapat mulai menelan data ke dalam pipa.

CLI

Gunakan get-pipeline-change-progressperintah untuk memeriksa status pipa. AWS CLI Permintaan berikut memeriksa status pipeline bernama`my-pipeline`:

```
aws osis get-pipeline-change-progress \
    --pipeline-name my-pipeline
```

Tanggapan:

```
{
```

```
   "ChangeProgressStatuses": {
      "ChangeProgressStages": [
         {
            "Description": "Validating pipeline configuration",
            "LastUpdated": 1.671055851E9,
            "Name": "VALIDATION",
            "Status": "PENDING"
         }
      ],
      "StartTime": 1.671055851E9,
      "Status": "PROCESSING",
      "TotalNumberOfStages": 5
   }
}
```

OpenSearch API Tertelan

Untuk melacak status pembuatan pipeline menggunakan API OpenSearch Ingestion, hubungi operasi. GetPipelineChangeProgress

Menggunakan cetak biru untuk membuat pipeline

Daripada membuat definisi pipeline dari awal, Anda dapat menggunakan cetak biru konfigurasi, yang merupakan templat YAMAL yang telah dikonfigurasi sebelumnya untuk skenario konsumsi umum seperti Trace Analytics atau log Apache. Cetak biru konfigurasi membantu Anda menyediakan saluran pipa dengan mudah tanpa harus membuat konfigurasi dari awal.

Konsol

Untuk menggunakan cetak biru pipa

1. Masuk ke konsol OpenSearch Layanan Amazon di https://console.aws.amazon.com/aos/home.
2. Pilih Pipelines di panel navigasi kiri dan pilih Create pipeline.
3. Di bawah Konfigurasi Pipeline, pilih Cetak biru Konfigurasi.
4. Pilih cetak biru. Konfigurasi pipeline diisi dengan sub-pipeline untuk kasus penggunaan yang Anda pilih.
5. Tinjau teks yang dikomentari yang memandu Anda melalui konfigurasi cetak biru.

> **Important**
>
> Cetak biru pipeline tidak valid apa adanya. Anda perlu membuat beberapa modifikasi, seperti menyediakan Wilayah AWS dan peran ARN untuk digunakan untuk otentikasi, jika tidak validasi pipeline akan gagal.

CLI

Untuk mendapatkan daftar semua cetak biru yang tersedia menggunakan AWS CLI, kirim permintaan. list-pipeline-blueprints

```
aws osis list-pipeline-blueprints
```

Permintaan mengembalikan daftar semua cetak biru yang tersedia.

Untuk mendapatkan informasi lebih rinci tentang cetak biru tertentu, gunakan perintah: get-pipeline-blueprint

```
aws osis get-pipeline-blueprint --blueprint-name AWS-ApacheLogPipeline
```

Permintaan ini mengembalikan isi cetak biru pipeline log Apache:

```
{
   "Blueprint":{
      "PipelineConfigurationBody":"###\n  # Limitations: https://docs.aws.amazon.com/
opensearch-service/latest/ingestion/ingestion.html#ingestion-limitations\n###\n###\n
 # apache-log-pipeline:\n    # This pipeline receives logs via http (e.g. FluentBit),
 extracts important values from the logs by matching\n    # the value in the 'log' key
 against the grok common Apache log pattern. The grokked logs are then sent\n    # to
 OpenSearch to an index named 'logs'\n###\n\nversion: \"2\"\napache-log-pipeline:\n
 source:\n    http:\n      # Provide the path for ingestion. ${pipelineName} will be
 replaced with pipeline name configured for this pipeline.\n      # In this case it
 would be \"/apache-log-pipeline/logs\". This will be the FluentBit output URI value.
\n      path: \"/${pipelineName}/logs\"\n  processor:\n    - grok:\n        match:\n
          log: [ \"%{COMMONAPACHELOG_DATATYPED}\" ]\n  sink:\n    - opensearch:\n
   # Provide an AWS OpenSearch Service domain endpoint\n        # hosts: [ \"https://
search-mydomain-1a2a3a4a5a6a7a8a9a0a9a8a7a.us-east-1.es.amazonaws.com\" ]\n
 aws:\n          # Provide a Role ARN with access to the domain. This role should have
 a trust relationship with osis-pipelines.amazonaws.com\n          # sts_role_arn:
```

```
 \"arn:aws:iam::123456789012:role/Example-Role\"\n          # Provide the region of the
 domain.\n          # region: \"us-east-1\"\n          # Enable the 'serverless' flag
 if the sink is an Amazon OpenSearch Serverless collection\n          # serverless:
 true\n        index: \"logs\"\n        # Enable the S3 DLQ to capture any failed
 requests in an S3 bucket\n        # dlq:\n          # s3:\n            # Provide an
 S3 bucket\n            # bucket: \"your-dlq-bucket-name\"\n            # Provide a key
 path prefix for the failed requests\n            # key_path_prefix: \"${pipelineName}/
logs/dlq\"\n            # Provide the region of the bucket.\n            # region:
 \"us-east-1\"\n            # Provide a Role ARN with access to the bucket. This role
 should have a trust relationship with osis-pipelines.amazonaws.com\n            #
 sts_role_arn: \"arn:aws:iam::123456789012:role/Example-Role\"\n",
      "BlueprintName":"AWS-ApacheLogPipeline"
    }
}
```

OpenSearch API Tertelan

Untuk mendapatkan informasi tentang cetak biru pipeline menggunakan API OpenSearch Ingestion, gunakan dan operasi. ListPipelineBlueprintsGetPipelineBlueprint

Melihat jaringan pipa Amazon OpenSearch Ingestion

Anda dapat melihat detail tentang pipeline Amazon OpenSearch Ingestion menggunakanAWS Management Console,AWS CLI, atau API OpenSearch Ingestion.

Konsol

Untuk melihat alur

1. Masuk ke konsol OpenSearch Layanan Amazon di https://console.aws.amazon.com/aos/home.
2. Pilih Alur di panel navigasi kiri.
3. (Opsional) Untuk melihat saluran pipa dengan status tertentu, pilih Status apa pun dan pilih status yang akan difilter.

   Alur dapat memiliki status berikut:

   - `Creating`— Alur sedang dibuat.
   - `Active`- Pipa aktif dan siap untuk menelan data.
   - `Updating`- Alur sedang diperbarui.
   - `Deleting`- Alur sedang dihapus.

- `Create failed`- Alur tidak dapat dibuat.
- `Update failed`- Pipa tidak dapat diperbarui.
- `Starting`- Pipa mulai.
- `Start failed`- Pipa tidak dapat dimulai.
- `Stopping`Alur sedang dihentikan.
- `Stopped`- Pipa dihentikan dan dapat dimulai ulang kapan saja.

Anda tidak ditagih untuk OCU Tertelan saat pipa berada di`Create failed`,,`Creating`, `Deleting` dan negara bagian. `Stopped`

CLI

Untuk melihat pipeline menggunakanAWS CLI, kirim permintaan list-pipelines:

```
aws osis list-pipelines
```

Permintaan mengembalikan daftar semua alur yang ada:

```
{
    "NextToken": null,
    "Pipelines": [
        {,
            "CreatedAt": 1.671055851E9,
            "LastUpdatedAt": 1.671055851E9,
            "MaxUnits": 4,
            "MinUnits": 2,
            "PipelineArn": "arn:aws:osis:us-west-2:123456789012:pipeline/log-pipeline",
            "PipelineName": "log-pipeline",
            "Status": "ACTIVE",
            "StatusReason": {
                "Description": "The pipeline is ready to ingest data."
            }
        },
            "CreatedAt": 1.671055851E9,
            "LastUpdatedAt": 1.671055851E9,
            "MaxUnits": 2,
            "MinUnits": 8,
            "PipelineArn": "arn:aws:osis:us-west-2:123456789012:pipeline/another-
pipeline",
```

```
            "PipelineName": "another-pipeline",
            "Status": "CREATING",
            "StatusReason": {
                "Description": "The pipeline is being created. It is not able to ingest
 data."
            }
        }
    ]
}
```

Untuk mendapatkan informasi tentang satu pipeline, gunakan perintah get-pipeline:

```
aws osis get-pipeline --pipeline-name "my-pipeline"
```

Permintaan mengembalikan informasi konfigurasi untuk pipeline yang ditentukan:

```
{
    "Pipeline": {
        "PipelineName": "my-pipeline",
        "PipelineArn": "arn:aws:osis:us-east-1:123456789012:pipeline/my-pipeline",
        "MinUnits": 9,
        "MaxUnits": 10,
        "Status": "ACTIVE",
        "StatusReason": {
            "Description": "The pipeline is ready to ingest data."
        },
        "PipelineConfigurationBody": "log-pipeline:\n source:\n http:\n processor:\n
 - grok:\n match:\nlog: [ '%{COMMONAPACHELOG}' ]\n - date:\n from_time_received: true
\n destination: \"@timestamp\"\n  sink:\n - opensearch:\n hosts: [ \"https://search-
mdp-performance-test-duxkb4qnycd63rpy6svmvyvfpi.us-east-1.es.amazonaws.com\" ]\n index:
 \"apache_logs\"\n aws_sts_role_arn: \"arn:aws:iam::123456789012:role/my-domain-role
\"\n  aws_region: \"us-east-1\"\n  aws_sigv4: true",,
        "CreatedAt": "2022-10-01T15:28:05+00:00",
        "LastUpdatedAt": "2022-10-21T21:41:08+00:00",
        "IngestEndpointUrls": [
            "my-pipeline-123456789012.us-east-1.osis.amazonaws.com"
        ]
    }
}
```

OpenSearchAPI Penyerapan

Untuk melihat pipeline OpenSearch Ingestion menggunakan API OpenSearch Ingestion, panggil dan operasinya. ListPipelinesGetPipeline

Memperbarui saluran pipa Amazon OpenSearch Ingestion

Anda dapat memperbarui pipeline Amazon OpenSearch Ingestion menggunakan, API AWS Management Console AWS CLI, atau Ingestion. OpenSearch OpenSearch Ingestion memulai penerapan biru/hijau saat Anda memperbarui konfigurasi YAMAL pipeline. Untuk informasi selengkapnya, lihat the section called “Penerapan biru/hijau untuk pembaruan saluran pipa”.

Topik

- Pertimbangan
- Izin diperlukan
- Memperbarui jaringan pipa
- Penerapan biru/hijau untuk pembaruan saluran pipa

Pertimbangan

Pertimbangkan hal berikut saat Anda memperbarui pipeline:

- Anda dapat mengedit batas kapasitas pipeline, opsi penerbitan log, dan konfigurasi YAMAL. Anda tidak dapat mengedit nama atau pengaturan jaringannya.
- Jika pipeline Anda menulis ke sink domain VPC, Anda tidak dapat kembali dan mengubah wastafel ke domain VPC yang berbeda setelah pipeline dibuat. Anda harus menghapus dan membuat ulang pipa dengan wastafel baru. Anda masih dapat mengalihkan sink dari domain VPC ke domain publik, dari domain publik ke domain VPC, atau dari domain publik ke domain publik lain.
- Anda dapat mengganti pipeline sink kapan saja antara domain OpenSearch Layanan publik dan koleksi OpenSearch Tanpa Server.
- Saat Anda memperbarui konfigurasi YAMAL pipeline, OpenSearch Ingestion memulai penerapan biru/hijau. Untuk informasi selengkapnya, lihat the section called “Penerapan biru/hijau untuk pembaruan saluran pipa”.
- Saat Anda memperbarui konfigurasi YAMAL pipeline, OpenSearch Ingestion akan secara otomatis memutakhirkan pipeline Anda ke versi minor terbaru yang didukung dari versi utama Penyediaan

Data yang ditentukan dalam konfigurasi pipeline. Proses ini membuat pipeline Anda tetap up to date dengan perbaikan bug terbaru dan peningkatan kinerja.

- Anda masih dapat melakukan pembaruan pada pipeline Anda saat dihentikan.

Izin diperlukan

OpenSearch Ingestion menggunakan izin IAM berikut untuk memperbarui pipeline:

- `osis:UpdatePipeline`— Perbarui pipa.
- `osis:ValidatePipeline`— Periksa apakah konfigurasi pipeline valid.
- `iam:PassRole`— Lewati peran pipeline ke OpenSearch Ingestion sehingga dapat menulis data ke domain. Izin ini hanya diperlukan jika Anda memperbarui konfigurasi YAMAL pipeline, bukan jika Anda memodifikasi setelan lain seperti penerbitan log atau batas kapasitas.

Misalnya, kebijakan berikut memberikan izin untuk memperbarui pipeline:

```
{
   "Version":"2012-10-17",
   "Statement":[
      {
         "Effect":"Allow",
         "Resource":"*",
         "Action":[
            "osis:UpdatePipeline",
            "osis:ValidatePipeline"
         ]
      },
      {
         "Resource":[
            "arn:aws:iam::{your-account-id}:role/{pipeline-role}"
         ],
         "Effect":"Allow",
         "Action":[
            "iam:PassRole"
         ]
      }
   ]
}
```

Memperbarui jaringan pipa

Anda dapat memperbarui pipeline Amazon OpenSearch Ingestion menggunakan, API AWS Management Console AWS CLI, atau Ingestion. OpenSearch

Konsol

Untuk memperbarui pipeline

1. Masuk ke konsol OpenSearch Layanan Amazon di https://console.aws.amazon.com/aos/home.
2. Pilih Pipelines di panel navigasi kiri.
3. Pilih pipeline untuk membuka pengaturannya. Anda dapat mengedit batas kapasitas pipeline, opsi penerbitan log, dan konfigurasi YAMAL. Anda tidak dapat mengedit nama atau pengaturan jaringannya.
4. Setelah selesai membuat perubahan, pilih Simpan.

CLI

Untuk memperbarui pipeline menggunakan AWS CLI, kirim permintaan update-pipeline. Contoh permintaan berikut mengunggah file konfigurasi baru dan memperbarui nilai kapasitas minimum dan maksimum:

```
aws osis update-pipeline \
  --pipeline-name "my-pipeline" \
  --pipline-configuration-body "file://new-pipeline-config.yaml" \
  --min-units 11 \
  --max-units 18
```

OpenSearch API Tertelan

Untuk memperbarui pipeline OpenSearch Ingestion menggunakan API OpenSearch Ingestion, panggil operasi. UpdatePipeline

Penerapan biru/hijau untuk pembaruan saluran pipa

OpenSearch Ingestion memulai proses penerapan biru/hijau saat Anda memperbarui konfigurasi YAMAL pipeline.

Biru/hijau mengacu pada praktik menciptakan lingkungan baru untuk pembaruan pipa dan perutean lalu lintas ke lingkungan baru setelah pembaruan tersebut selesai. Praktik ini meminimalkan waktu

henti dan mempertahankan lingkungan asli jika deployment ke lingkungan baru tidak berhasil. Penerapan biru/hijau sendiri tidak memiliki dampak kinerja apa pun, tetapi kinerja mungkin berubah jika konfigurasi pipeline Anda berubah dengan cara yang mengubah kinerja.

OpenSearch Penyerapan memblokir auto-scaling selama penerapan biru/hijau. Anda terus dikenakan biaya hanya untuk lalu lintas ke pipa lama sampai diarahkan ke pipa baru. Setelah lalu lintas dialihkan, Anda hanya dikenakan biaya untuk pipeline baru. Anda tidak pernah dikenakan biaya untuk dua saluran pipa secara bersamaan.

Saat Anda memperbarui file konfigurasi YAMAL pipeline, OpenSearch Ingestion dapat secara otomatis memutakhirkan pipeline Anda ke versi minor terbaru yang didukung dari versi utama Penyediaan Data yang ditentukan dalam konfigurasi pipeline. Misalnya, Anda mungkin memiliki `version: "2"` konfigurasi pipeline, dan OpenSearch Ingestion awalnya menyediakan pipeline dengan versi 2.1.0. Saat dukungan untuk versi 2.1.1 ditambahkan, dan Anda membuat perubahan pada konfigurasi pipeline, OpenSearch Ingestion akan meningkatkan pipeline Anda ke versi 2.1.1.

Proses ini membuat pipeline Anda tetap up to date dengan perbaikan bug terbaru dan peningkatan kinerja. OpenSearch Ingestion tidak dapat memperbarui versi utama pipeline Anda kecuali Anda mengubah `version` opsi secara manual dalam konfigurasi pipeline.

Menghentikan dan memulai jaringan pipa Amazon OpenSearch Ingestion

Menghentikan dan memulai alur AmazonOpenSearch. Anda dapat menghentikan alur untuk sementara dan merombakan alur.

Topik

- Ikhtisar menghentikan dan OpenSearch memulai alur
- Menghentikan pipeline OpenSearch Ingestion
- Memulai pipeline OpenSearch Ingestion

Ikhtisar menghentikan dan OpenSearch memulai alur

Anda dapat menghentikan pipeline selama periode di mana Anda tidak perlu memasukkan data ke dalamnya. Anda dapat memulai alur lagi kapan saja Anda memerlukannya. Memulai dan menghentikan menyederhanakan proses pengaturan dan perombakan pada alur yang digunakan

untuk pengembangan, pengujian, atau aktifitas serupa yang tidak memerlukan ketersediaan berkelanjutan.

Saat pipeline Anda dihentikan, Anda tidak dikenakan biaya untuk setiap jam OCU Tertelan. Anda masih dapat memperbarui saluran pipa yang berhenti, dan mereka menerima pembaruan versi minor otomatis dan patch keamanan.

Jangan gunakan memulai dan menghentikan jika Anda perlu menjaga alur Anda tetap berjalan tetapi memiliki lebih banyak kapasitas yang Anda butuhkan. Jika pipa Anda terlalu mahal atau tidak terlalu sibuk, pertimbangkan untuk mengurangi batas kapasitas maksimumnya. Untuk informasi selengkapnya, lihat the section called “Penskalaan pipa”.

Menghentikan pipeline OpenSearch Ingestion

Untuk menggunakan pipa OpenSearch Ingestion atau melakukan administrasi, Anda selalu mulai dengan pipa aktif, kemudian menghentikan pipa, dan kemudian mulai pipa lagi. Sementara saluran pipa Anda dihentikan, Anda tidak dikenakan biaya untuk jam Menelan OCU.

Konsol

Untuk menghentikan

1. Masuk ke konsol OpenSearch Layanan Amazon di https://console.aws.amazon.com/aos/home.
2. Di panel navigasi, pilih Alur, lalu pilih alur. Anda dapat melakukan operasi penghentian dari halaman ini, atau navigasi ke halaman detail untuk alur yang ingin Anda hentikan.
3. Untuk Tindakan, pilih Stop pipeline.

   Jika pipeline tidak dapat dihentikan dan dimulai, tindakan Stop pipeline tidak tersedia.

AWS CLI

Untuk menghentikan alurAWS CLI, panggil perintah stop-alur dengan parameter berikut ini:

- `--pipeline-name`- Nama dari alur.

Example

```
aws osis stop-pipeline --pipeline-name my-pipeline
```

OpenSearchPenyerapan

Untuk menghentikan pipeline menggunakan API OpenSearch Ingestion, panggil StopPipelineoperasi dengan parameter berikut:

- `PipelineName`- Nama dari alur.

Memulai pipeline OpenSearch Ingestion

Anda selalu memulai OpenSearch Alur Penyerapan yang memulai dengan alur yang sudah berada dalam status terhenti. Alur menjaga pengaturan seperti batas kapasitas, pengaturan jaringan, dan opsi penerbitan log.

Memulai alur biasanya memerlukan waktu beberapa menit.

Konsol

Untuk memulai

1. Masuk ke konsol OpenSearch Layanan Amazon di https://console.aws.amazon.com/aos/home.
2. Di panel navigasi, pilih Alur, lalu pilih alur. Anda dapat melakukan operasi mulai dari halaman ini, atau navigasi ke halaman detail untuk alur yang ingin Anda mulai.
3. Untuk Tindakan, pilih Mulai pipeline.

AWS CLI

Untuk memulai alur dengan menggunakanAWS CLI, panggil perintah start-alur dengan parameter berikut ini:

- `--pipeline-name`- Nama dari alur.

Example

```
aws osis start-pipeline --pipeline-name my-pipeline
```

OpenSearchPenyerapan

Untuk memulai pipeline OpenSearch Ingestion menggunakan API OpenSearch Ingestion, panggil StartPipelineoperasi dengan parameter berikut:

- `PipelineName`- Nama dari alur.

Menghapus jaringan pipa Amazon OpenSearch Ingestion

Anda dapat menghapus pipeline Amazon OpenSearch Ingestion menggunakanAWS Management Console,AWS CLI, atau API OpenSearch Ingestion. Anda tidak dapat menghapus pipeline bila memiliki status `Creating` atau`Updating`.

Konsol

Untuk menghapus Alur

1. Masuk ke konsol Amazon OpenSearch Service di https://console.aws.amazon.com/aos/home.
2. Pilih Alur di sebelah kiri.
3. Pilih Alur yang ingin Anda hapus.
4. Konfirmasikan penghapusan dan pilih Hapus.

CLI

Untuk menghapus pipeline menggunakanAWS CLI, kirim permintaan delete-pipeline:

```
aws osis delete-pipeline --pipeline-name "my-pipeline"
```

OpenSearchAPI Penyerapan

Untuk menghapus pipeline OpenSearch Ingestion menggunakan API OpenSearch Ingestion, panggil DeletePipelineoperasi dengan parameter berikut:

- `PipelineName`- Nama dari lu.

Plugin dan opsi yang didukung untuk saluran Amazon OpenSearch Ingestion

Amazon OpenSearch Ingestion mendukung subset sumber, prosesor, dan sink dibandingkan dengan open source Data Prepper. Selain itu, ada beberapa kendala yang Ditempatkan oleh OpenSearch

Ingestion pada opsi yang tersedia untuk setiap plugin yang didukung. Bagian berikut menjelaskan plugin dan opsi terkait yang didukung oleh OpenSearch Ingestion.

> **Note**
>
> OpenSearch Ingestion tidak mendukung plugin buffer apa pun karena secara otomatis mengkonfigurasi buffer default. Anda menerima kesalahan validasi jika menyertakan buffer dalam konfigurasi pipeline Anda.

Topik

Plugin yang didukung

OpenSearch Ingestion mendukung plugin Data Prepper berikut:

Sumber:

- Dynamodb
- OpenSearch

- HTTP
- Kafka
- Log oTel
- Metrik OTel
- Jejak oTel
- S3

Prosesor:

- Agregat
- Detektor anomali

- CSV
- Tanggal
- Dekompresi
- Membedah
- Jatuhkan acara
- Geo IP
- Grok
- Nilai kunci
- Peta ke daftar
- Mutasi peristiwa (serangkaian prosesor)
- Mutasi string (serangkaian prosesor)
- Mengaburkan
- Metrik OTel
- Grup jejak oTel
- Jejak oTel
- Parse Ion
- Mengurai JSON
- Mengurai XML
- Pilih entri
- Peta layanan
- Lacak peer forwarder
- Memangkas
- Agen pengguna

Wastafel:

- OpenSearch(mendukung OpenSearch Layanan, OpenSearch Tanpa Server, dan Elasticsearch 6.8 atau yang lebih baru)
- S3

Codec wastafel:

- Avro
- NDJSON
- JSON
- Parket

Prosesor stateless versus stateful

Prosesor stateless melakukan operasi seperti transformasi dan penyaringan, sementara prosesor stateful melakukan operasi seperti agregasi, yang mengingat hasil dari proses sebelumnya. OpenSearch Ingestion mendukung prosesor stateful Agregate dan Service-map. Semua prosesor lain yang didukung adalah stateless.

Untuk jaringan pipa yang hanya berisi prosesor stateless, batas kapasitas maksimum adalah 96 OCU konsumsi. Jika pipa berisi prosesor stateful, batas kapasitas maksimum adalah 48 OCU konsumsi. Namun, jika pipeline memiliki buffering persisten yang diaktifkan, ia dapat memiliki maksimum 384 OCU Ingestion dengan hanya prosesor stateless, atau 192 OCU Ingestion jika berisi prosesor stateful. Untuk informasi selengkapnya, lihat the section called “Penskalaan pipa”.

nd-to-end Pengakuan E hanya didukung untuk prosesor stateless. Untuk informasi selengkapnya, lihat the section called “E nd-to-end pengakuan”.

Persyaratan dan kendala konfigurasi

Kecuali ditentukan lain di bawah ini, semua opsi yang dijelaskan dalam referensi konfigurasi Persiapan Data untuk plugin yang didukung yang tercantum di atas diizinkan dalam pipeline OpenSearch Ingestion. Bagian berikut menjelaskan kendala yang Ditempatkan OpenSearch Ingestion pada opsi plugin tertentu.

> **Note**
>
> OpenSearch Ingestion tidak mendukung plugin buffer apa pun karena secara otomatis mengkonfigurasi buffer default. Anda menerima kesalahan validasi jika menyertakan buffer dalam konfigurasi pipeline Anda.

Banyak opsi dikonfigurasi dan dikelola secara internal oleh OpenSearch Ingestion, seperti dan. `authentication` `acm_certificate_arn` Opsi lain, seperti `thread_count`

dan`request_timeout`, memiliki dampak kinerja jika diubah secara manual. Oleh karena itu, nilai-nilai ini ditetapkan secara internal untuk memastikan kinerja pipa Anda yang optimal.

Terakhir, beberapa opsi tidak dapat diteruskan ke OpenSearch Ingestion, seperti `ism_policy_file` dan`sink_template`, karena mereka adalah file lokal ketika dijalankan di Prepper Data sumber terbuka. Nilai-nilai ini tidak didukung.

Topik

Opsi pipa umum

Opsi pipeline umum berikut disetel oleh OpenSearch Ingestion dan tidak didukung dalam konfigurasi pipeline:

- `workers`
- `delay`

Prosesor Grok

Opsi prosesor Grok berikut tidak didukung:

- `patterns_directories`
- `patterns_files_glob`

Sumber HTTP

Plugin sumber HTTP memiliki persyaratan dan kendala berikut:

- `path`Opsi ini diperlukan. Path adalah string seperti`/log/ingest`, yang mewakili jalur URI untuk log ingestion. Path ini mendefinisikan URI yang Anda gunakan untuk mengirim data ke pipeline. Misalnya, `https://log-pipeline.us-west-2.osis.amazonaws.com`*`/log/ingest`*. Jalur harus dimulai dengan garis miring (/), dan dapat berisi karakter khusus '-', '_', ' . ', dan'/', serta `${pipelineName}` placeholder.
- Opsi sumber HTTP berikut disetel oleh OpenSearch Ingestion dan tidak didukung dalam konfigurasi pipeline:
  - `port`
  - `ssl`
  - `ssl_key_file`
  - `ssl_certificate_file`
  - `aws_region`
  - `authentication`
  - `unauthenticated_health_check`
  - `use_acm_certificate_for_ssl`
  - `thread_count`
  - `request_timeout`
  - `max_connection_count`
  - `max_pending_requests`
  - `health_check_service`
  - `acm_private_key_password`
  - `acm_certificate_timeout_millis`
  - `acm_certificate_arn`

OpenSearch wastafel

Plugin OpenSearchwastafel memiliki persyaratan dan batasan berikut.

- `aws`Opsi ini diperlukan, dan harus berisi opsi berikut:
  - `sts_role_arn`
  - `region`
  - `hosts`

  - `serverless`(jika wastafel adalah koleksi OpenSearch Tanpa Server)
- `sts_role_arn`Opsi harus menunjuk ke peran yang sama untuk setiap sink dalam file definisi YAMAL.
- `hosts`Opsi harus menentukan titik akhir domain OpenSearch Layanan atau titik akhir koleksi OpenSearch Tanpa Server. Semua host dalam file definisi YAMAL harus menunjuk ke titik akhir yang sama. Anda tidak dapat menentukan titik akhir kustom untuk domain; itu harus menjadi titik akhir standar.
- Jika `hosts` opsi adalah titik akhir koleksi tanpa server, Anda harus mengatur opsi ke. `serverless` `true` Selain itu, jika file definisi YAMAL Anda berisi `index_type` opsi, itu harus disetel kemanagement_disabled, jika tidak validasi gagal.
- Opsi berikut tidak didukung:
  - `username`
  - `password`
  - `cert`
  - `proxy`
  - `dlq_file`- Jika Anda ingin membongkar peristiwa gagal ke antrian huruf mati (DLQ), Anda harus menggunakan `dlq` opsi dan menentukan ember S3.
  - `ism_policy_file`
  - `socket_timeout`
  - `template_file`
  - `insecure`
  - `bulk_size`

Sumber metrik OTel, sumber jejak OTel, dan sumber log OTel

Sumber metrik OTel, sumber jejak OTel, dan plugin sumber log OTel memiliki persyaratan dan batasan berikut:

- `path`Opsi ini diperlukan. Path adalah string seperti`/log/ingest`, yang mewakili jalur URI untuk log ingestion. Path ini mendefinisikan URI yang Anda gunakan untuk mengirim data ke pipeline. Misalnya, `https://log-pipeline.us-west-2.osis.amazonaws.com`*`/log/ingest`*. Jalur harus dimulai dengan garis miring (/), dan dapat berisi karakter khusus '-', '_', ' . ', dan'/', serta `${pipelineName}` placeholder.

- Opsi berikut disetel oleh OpenSearch Ingestion dan tidak didukung dalam konfigurasi pipeline:
  - `port`
  - `ssl`
  - `sslKeyFile`
  - `sslKeyCertChainFile`
  - `authentication`
  - `unauthenticated_health_check`
  - `useAcmCertForSSL`
  - `unframed_requests`
  - `proto_reflection_service`
  - `thread_count`
  - `request_timeout`
  - `max_connection_count`
  - `acmPrivateKeyPassword`
  - `acmCertIssueTimeOutMillis`
  - `health_check_service`
  - `acmCertificateArn`
  - `awsRegion`

Prosesor grup jejak OTel

Prosesor grup jejak OTel memiliki persyaratan dan batasan berikut:

- `aws`Opsi ini diperlukan, dan harus berisi opsi berikut:
  - `sts_role_arn`
  - `region`
  - `hosts`
- `sts_role_arn`Opsi menentukan peran yang sama dengan peran pipeline yang Anda tentukan dalam konfigurasi OpenSearch wastafel.
- `insecure`Opsi `username``password`,`cert`, dan tidak didukung.
- `aws_sigv4`Opsi ini diperlukan dan harus disetel ke true.

- `serverless`Opsi dalam plugin OpenSearch sink tidak didukung. Prosesor grup jejak Otel saat ini tidak berfungsi dengan koleksi Tanpa OpenSearch Server.
- Jumlah `otel_trace_group` prosesor dalam badan konfigurasi pipa tidak boleh melebihi 8.

Prosesor jejak OTel

Prosesor jejak OTel memiliki persyaratan dan batasan berikut:

- Nilai `trace_flush_interval` opsi tidak boleh melebihi 300 detik.

Prosesor peta layanan

Prosesor peta layanan memiliki persyaratan dan batasan berikut:

- Nilai `window_duration` opsi tidak boleh melebihi 300 detik.

Sumber S3

Plugin sumber S3 memiliki persyaratan dan batasan berikut:

- `aws`Opsi ini diperlukan, dan harus berisi `region` dan `sts_role_arn` opsi.
- Nilai `records_to_accumulate` opsi tidak boleh melebihi 200.
- Nilai `maximum_messages` opsi tidak boleh melebihi 10.
- Jika ditentukan, `disable_bucket_ownership_validation` opsi harus diatur ke false.
- Jika ditentukan, `input_serialization` opsi harus diatur ke`parquet`.

Bekerja dengan integrasi pipa Amazon OpenSearch Ingestion

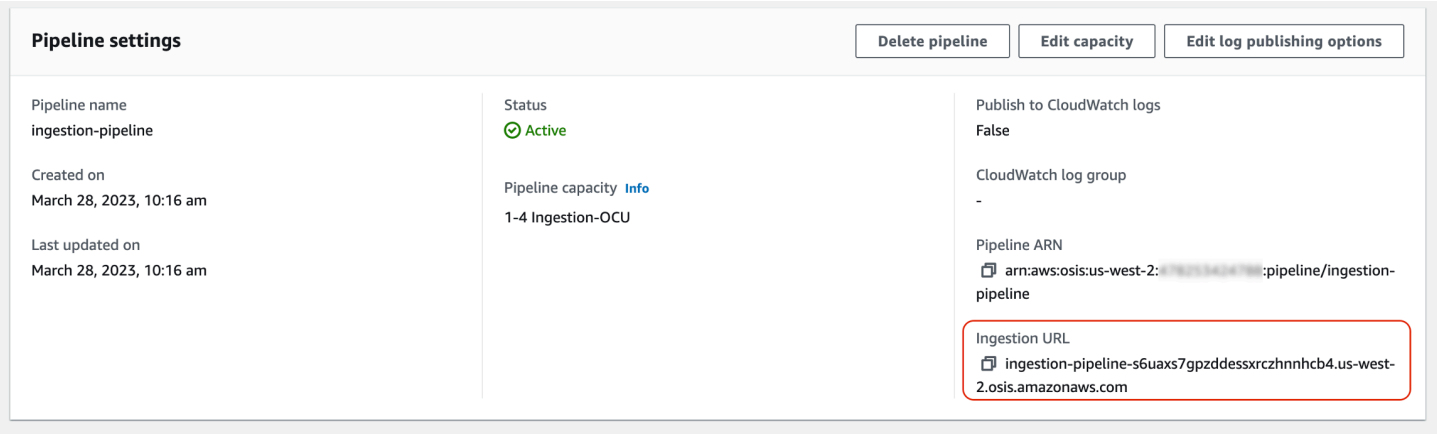
Agar berhasil menyerap data ke dalam pipeline Amazon OpenSearch Ingestion, Anda harus mengonfigurasi aplikasi klien Anda (sumber) untuk mengirim data ke titik akhir pipeline. Sumber Anda mungkin klien seperti log Fluent Bit, OpenTelemetry Collector, atau bucket S3 sederhana. Konfigurasi yang tepat berbeda untuk setiap klien.

Perbedaan penting selama konfigurasi sumber (dibandingkan dengan mengirim data langsung ke domain OpenSearch Layanan atau koleksi OpenSearch Tanpa Server) adalah nama AWS layanan (`osis`) dan titik akhir host, yang harus menjadi titik akhir pipeline.

Topik

Membangun titik akhir konsumsi

Untuk menyerap data ke dalam pipeline, kirimkan ke titik akhir konsumsi. Untuk menemukan URL konsumsi, navigasikan ke halaman pengaturan Pipeline dan salin URL Ingestion:



Untuk membangun titik akhir konsumsi penuh untuk sumber berbasis tarik seperti jejak OTel dan metrik OTel, tambahkan jalur konsumsi dari konfigurasi pipeline Anda ke URL konsumsi.

Misalnya, katakan bahwa konfigurasi pipeline Anda memiliki jalur konsumsi berikut:

```
entry-pipeline:
  source:
    http:
      path: "/my/test_path"
```

Titik akhir konsumsi penuh, yang Anda tentukan dalam konfigurasi klien Anda, akan mengambil format berikut: `https://ingestion-pipeline-abcdefg.us-west-2.osis.amazonaws.com/my/test_path`

Untuk informasi selengkapnya, lihat the section called “Menentukan jalur konsumsi”.

Membuat peran konsumsi

Semua permintaan untuk OpenSearch Ingestion harus ditandatangani dengan Signature Version 4. Minimal, peran yang menandatangani permintaan harus diberikan izin untuk `osis:Ingest` tindakan tersebut, yang memungkinkannya mengirim data ke pipa OpenSearch Ingestion.

Misalnya, kebijakan AWS Identity and Access Management (IAM) berikut memungkinkan peran terkait untuk mengirim data ke satu pipeline:

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Effect": "Allow",
      "Action": "osis:Ingest",
      "Resource": "arn:aws:osis:us-east-1:{account-id}:pipeline/pipeline-name"
    }
  ]
}
```

> **Note**
>
> Untuk menggunakan peran untuk semua pipeline, ganti ARN di `Resource` elemen dengan wildcard (*).

Menyediakan akses konsumsi lintas akun

> **Note**
>
> Anda hanya dapat menyediakan akses konsumsi lintas akun untuk saluran pipa publik, bukan saluran pipa VPC.

Anda mungkin perlu memasukkan data ke dalam pipeline dari yang lain Akun AWS, seperti akun yang menampung aplikasi sumber Anda. Jika prinsipal yang menulis ke pipeline berada di akun yang berbeda dari pipeline itu sendiri, Anda perlu mengonfigurasi prinsipal untuk mempercayai peran IAM lain untuk menyerap data ke dalam pipeline.

Untuk mengonfigurasi izin konsumsi lintas akun

1. Buat peran konsumsi dengan `osis:Ingest` izin (dijelaskan di bagian sebelumnya) dalam hal yang Akun AWS sama dengan pipeline. Untuk petunjuk, lihat Membuat peran IAM.
2. Lampirkan kebijakan kepercayaan ke peran konsumsi yang memungkinkan kepala sekolah di akun lain untuk menganggapnya:

```
{
  "Version": "2012-10-17",
  "Statement": [{
     "Effect": "Allow",
     "Principal": {
       "AWS": "arn:aws:iam::{external-account-id}:root"
      },
     "Action": "sts:AssumeRole"
  }]
}
```

3. Di akun lain, konfigurasikan aplikasi klien Anda (misalnya, Fluent Bit) untuk mengambil peran konsumsi. Agar ini berfungsi, akun aplikasi harus memberikan izin kepada pengguna aplikasi atau peran untuk mengambil peran konsumsi.

   Contoh kebijakan berbasis identitas berikut memungkinkan prinsipal terlampir untuk berasumsi `ingestion-role` dari akun pipeline:

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Effect": "Allow",
      "Action": "sts:AssumeRole",
      "Resource": "arn:aws:iam::{account-id}:role/ingestion-role"
    }
  ]
}
```

Aplikasi klien kemudian dapat menggunakan AssumeRoleoperasi untuk mengasumsikan `ingestion-role` dan menyerap data ke dalam pipeline terkait.

Menggunakan pipeline OpenSearch Ingestion dengan Amazon DynamoDB

Anda dapat menggunakan pipeline OpenSearch Ingestion dengan DynamoDB untuk mengalirkan peristiwa tabel DynamoDB (seperti membuat, memperbarui, dan menghapus) ke domain dan koleksi Amazon Service. OpenSearch Pipeline OpenSearch Ingestion menggabungkan infrastruktur change data capture (CDC) untuk menyediakan cara berskala tinggi dan latensi rendah untuk terus mengalirkan data dari tabel DynamoDB.

Ada dua cara Anda dapat menggunakan DynamoDB sebagai sumber untuk memproses data—dengan dan tanpa snapshot awal penuh.

Snapshot awal lengkap adalah cadangan tabel yang diambil DynamoDB dengan point-in-time fitur pemulihan (PITR). DynamoDB mengunggah snapshot ini ke Amazon S3. Dari sana, pipa OpenSearch Ingestion mengirimkannya ke satu indeks dalam domain, atau mempartisinya ke beberapa indeks dalam domain. Untuk menjaga data di DynamoDB OpenSearch dan konsisten, pipeline menyinkronkan semua peristiwa buat, perbarui, dan hapus di tabel DynamoDB dengan dokumen yang disimpan dalam indeks atau indeks. OpenSearch

Saat Anda menggunakan snapshot awal penuh, pipeline OpenSearch Ingestion Anda pertama-tama akan menyerap snapshot dan kemudian mulai membaca data dari DynamoDB Streams. Ini akhirnya mengejar dan mempertahankan konsistensi data hampir real-time antara DynamoDB dan. OpenSearch Ketika Anda memilih opsi ini, Anda harus mengaktifkan PITR dan aliran DynamoDB di meja Anda.

Anda juga dapat menggunakan integrasi OpenSearch Ingestion dengan DynamoDB untuk melakukan streaming peristiwa tanpa snapshot. Pilih opsi ini jika Anda sudah memiliki snapshot lengkap dari beberapa mekanisme lain, atau jika Anda hanya ingin melakukan streaming peristiwa saat ini dari tabel DynamoDB dengan DynamoDB Streams. Ketika Anda memilih opsi ini, Anda hanya perlu mengaktifkan aliran DynamoDB di meja Anda.

Untuk informasi selengkapnya tentang integrasi ini, lihat Integrasi DynamoDB Zero-ETL dengan OpenSearch Amazon Service di Panduan Pengembang.Amazon DynamoDB

Topik

- Prasyarat
- Langkah 1: Konfigurasikan peran pipeline

- Langkah 2: Buat pipa
- Konsistensi data
- Pemetaan tipe data
- Batasan

Prasyarat

Untuk mengatur pipeline, Anda harus memiliki tabel DynamoDB dengan DynamoDB Streams diaktifkan. Streaming Anda harus menggunakan jenis tampilan `NEW_IMAGE` aliran. Namun, saluran pipa OpenSearch Ingestion juga dapat mengalirkan peristiwa `NEW_AND_OLD_IMAGES` jika jenis tampilan aliran ini sesuai dengan kasus penggunaan Anda.

Jika Anda menggunakan snapshot, Anda juga harus mengaktifkan point-in-time pemulihan di meja Anda. Untuk informasi selengkapnya, lihat Membuat tabel, Mengaktifkan point-in-time pemulihan, dan Mengaktifkan aliran di Panduan Pengembang Amazon DynamoDB.

Langkah 1: Konfigurasikan peran pipeline

Setelah tabel DynamoDB disiapkan, siapkan peran pipeline yang ingin Anda gunakan dalam konfigurasi pipeline, dan tambahkan izin DynamoDB berikut dalam peran:

```
{
    "Version": "2012-10-17",
    "Statement": [
        {
            "Sid": "allowRunExportJob",
            "Effect": "Allow",
            "Action": [
                "dynamodb:DescribeTable",
                "dynamodb:DescribeContinuousBackups",
                "dynamodb:ExportTableToPointInTime"
            ],
            "Resource": [
                "arn:aws:dynamodb:us-east-1:{account-id}:table/my-table"
            ]
        },
        {
            "Sid": "allowCheckExportjob",
            "Effect": "Allow",
            "Action": [
```

```
                "dynamodb:DescribeExport"
            ],
            "Resource": [
                "arn:aws:dynamodb:us-east-1:{account-id}:table/my-table/export/*"
            ]
        },
        {
            "Sid": "allowReadFromStream",
            "Effect": "Allow",
            "Action": [
                "dynamodb:DescribeStream",
                "dynamodb:GetRecords",
                "dynamodb:GetShardIterator"
            ],
            "Resource": [
                "arn:aws:dynamodb:us-east-1:{account-id}:table/my-table/stream/*"
            ]
        },
        {
            "Sid": "allowReadAndWriteToS3ForExport",
            "Effect": "Allow",
            "Action": [
                "s3:GetObject",
                "s3:AbortMultipartUpload",
                "s3:PutObject",
                "s3:PutObjectAcl"
            ],
            "Resource": [
                "arn:aws:s3:::my-bucket/export/*"
            ]
        }
    ]
}
```

Anda juga dapat menggunakan kunci yang dikelola AWS KMS pelanggan untuk mengenkripsi file data ekspor. Untuk mendekripsi objek yang diekspor, tentukan `s3_sse_kms_key_id` ID kunci dalam konfigurasi ekspor pipa dengan format berikut:. `arn:aws:kms:us-west-2:{account-id}:key/my-key-id`

Langkah 2: Buat pipa

Anda kemudian dapat mengonfigurasi pipeline OpenSearch Ingestion seperti berikut ini, yang menentukan DynamoDB sebagai sumbernya. Pipeline sampel ini menyerap data dari `table-a`

snapshot PITR, diikuti oleh peristiwa dari DynamoDB Streams. Posisi awal `LATEST` menunjukkan bahwa pipeline harus membaca data terbaru dari DynamoDB Streams.

```
version: "2"
cdc-pipeline:
  source:
    dynamodb:
      tables:
      - table_arn: "arn:aws:dynamodb:us-west-2:{account-id}:table/table-a"
        export:
          s3_bucket: "my-bucket"
          s3_prefix: "export/"
        stream:
          start_position: "LATEST"
      aws:
        region: "us-west-2"
        sts_role_arn: "arn:aws:iam::{account-id}:role/pipeline-role"
  sink:
  - opensearch:
      hosts: ["https://search-mydomain.us-east-1.es.amazonaws.com"]
      index: "${getMetadata(\"table_name\")}"
      index_type: custom
      document_id: "${getMetadata(\"primary_key\")}"
      action: "${getMetadata(\"opensearch_action\")}"
      document_version: "${getMetadata(\"document_version\")}"
      document_version_type: "external"
```

Anda dapat menggunakan cetak biru AWS-DynamoDB ChangeDataCapturePipeline atau SingleTableDesignPipelineAWS-DynamoDB untuk membuat pipeline ini. Untuk informasi selengkapnya, lihat the section called “Menggunakan cetak biru untuk membuat pipeline”.

Konsistensi data

OpenSearch Ingestion mendukung end-to-end pengakuan untuk memastikan daya tahan data. Ketika pipeline membaca snapshot atau stream, ia secara dinamis membuat partisi untuk pemrosesan paralel. Pipeline menandai partisi sebagai lengkap ketika menerima pengakuan setelah menelan semua catatan dalam OpenSearch domain atau koleksi.

Jika Anda ingin memasukkan ke dalam koleksi pencarian OpenSearch Tanpa Server, Anda dapat membuat ID dokumen di pipeline. Jika Anda ingin memasukkan koleksi deret waktu OpenSearch Tanpa Server, perhatikan bahwa pipeline tidak menghasilkan ID dokumen.

Pipeline OpenSearch Ingestion juga memetakan tindakan peristiwa yang masuk ke dalam tindakan pengindeksan massal yang sesuai untuk membantu menyerap dokumen. Ini membuat data tetap konsisten, sehingga setiap perubahan data di DynamoDB direkonsiliasi dengan perubahan dokumen yang sesuai. OpenSearch

Pemetaan tipe data

OpenSearch Layanan secara dinamis memetakan tipe data di setiap dokumen yang masuk ke tipe data yang sesuai di DynamoDB. Tabel berikut menunjukkan bagaimana OpenSearch Layanan secara otomatis memetakan berbagai tipe data.

Tipe data	OpenSearch	DynamoDB
Angka	OpenSearch secara otomatis memetakan data numerik. Jika angka tersebut adalah bilangan bulat, OpenSearch petakan sebagai nilai panjang. Jika angkanya pecahan, maka OpenSearch petakan sebagai nilai float. OpenSearch secara dinamis memetakan berbagai atribut berdasarkan dokumen terkirim pertama. Jika Anda memiliki campuran tipe data untuk atribut yang sama di DynamoDB, seperti bilangan bulat dan bilangan pecahan, pemetaan mungkin gagal. Misalnya, jika dokumen pertama Anda memiliki atribut yang merupakan bilangan bulat, dan dokumen selanjutn ya memiliki atribut yang sama dengan angka pecahan, OpenSearch gagal untuk menelan dokumen kedua. Dalam kasus ini, Anda harus menyediakan template pemetaan eksplisit, seperti berikut ini:	DynamoDB mendukung angka.

Tipe data	OpenSearch	DynamoDB

	<pre>{
	"template": {
	"mappings": {
	"properties": {
	"MixedNumberAttribute": {
	"type": "float"
	}
	}
	}
	}
	}</pre>
	Jika Anda membutuhkan presisi ganda, gunakan pemetaan bidang tipe string. Tidak ada tipe numerik setara yang mendukung 38 digit presisi dalam. OpenSearch	

Jumlah set	OpenSearch secara otomatis memetakan angka yang ditetapkan ke dalam array baik nilai panjang atau nilai float. Seperti halnya bilangan skalar, ini tergantung pada apakah angka pertama yang dicerna adalah bilangan bulat atau bilangan pecahan. Anda dapat memberikan pemetaan untuk kumpulan angka dengan cara yang sama seperti Anda memetakan string skalar.	DynamoDB mendukung jenis yang mewakili set angka.

Tipe data	OpenSearch	DynamoDB
String	OpenSearch secara otomatis memetakan nilai string sebagai teks. Dalam beberapa situasi, seperti nilai yang disebutkan, Anda dapat memetakan ke jenis kata kunci. Contoh berikut menunjukkan bagaimana memetakan atribut DynamoDB `PartType` bernama untuk kata kunci. OpenSearch `{ "template": { "mappings": { "properties": { "PartType": { "type": "keyword" } } } } }`	DynamoDB mendukung string.
Set string	OpenSearch secara otomatis memetakan string yang diatur ke dalam array string. Anda dapat memberikan pemetaan untuk set string dengan cara yang sama seperti Anda memetakan string skalar.	DynamoDB mendukung jenis yang mewakili set string.

Tipe data	OpenSearch	DynamoDB
Biner	OpenSearch secara otomatis memetakan data biner sebagai teks. Anda dapat memberikan pemetaan untuk menulis ini sebagai bidang OpenSearch biner. Contoh berikut menunjukkan bagaimana memetakan atribut DynamoDB `ImageData` bernama ke OpenSearch bidang biner. `{ "template": { "mappings": { "properties": { "ImageData": { "type": "binary" } } } } }`	DynamoDB mendukung atribut tipe biner.
Set biner	OpenSearch secara otomatis memetakan set biner ke dalam array data biner sebagai teks. Anda dapat memberikan pemetaan untuk kumpulan angka dengan cara yang sama seperti Anda memetakan biner skalar.	DynamoDB mendukung jenis yang mewakili set nilai biner.
Boolean	OpenSearch memetakan tipe DynamoDB Boolean ke dalam tipe Boolean. OpenSearch	DynamoDB mendukung atribut tipe Boolean.

Tipe data	OpenSearch	DynamoDB
Null	OpenSearch dapat menelan dokumen dengan tipe nol DynamoDB. Ini menyimpan nilai sebagai nilai nol dalam dokumen. Tidak ada pemetaan untuk jenis ini, dan bidang ini tidak diindeks atau dicari. Jika nama atribut yang sama digunakan untuk tipe null dan kemudian berubah ke tipe yang berbeda seperti string, OpenSearch membuat pemetaan dinamis untuk nilai non-null pertama. Nilai selanjutnya masih bisa berupa nilai nol DynamoDB.	DynamoDB mendukung atribut tipe null.

Tipe data	OpenSearch	DynamoDB
Peta	OpenSearch memetakan atribut peta DynamoDB ke bidang bersarang. Pemetaan yang sama berlaku dalam bidang bersarang. Contoh berikut memetakan string dalam bidang bersarang ke jenis kata kunci di OpenSearch: `{ "template": { "mappings": { "properties": { "AdditionalDescriptions": { "properties": { "PartType": { "type": "keyword" } } } } } } }`	DynamoDB mendukung atribut tipe peta.

Tipe data	OpenSearch	DynamoDB
Daftar	OpenSearch memberikan hasil yang berbeda untuk daftar DynamoDB, tergantung pada apa yang ada dalam daftar. Ketika daftar berisi semua jenis jenis skalar yang sama (misalnya, daftar semua string), kemudian OpenSearc h mencerna daftar sebagai array dari jenis itu. Ini berfungsi untuk tipe string, number, Boolean, dan null. Pembatasa n untuk masing-masing jenis ini sama dengan batasan untuk skalar jenis itu. Anda juga dapat menyediakan pemetaan untuk daftar peta dengan menggunakan pemetaan yang sama seperti yang akan Anda gunakan untuk peta. Anda tidak dapat memberikan daftar tipe campuran.	DynamoDB mendukung atribut tipe daftar.

Tipe data	OpenSearch	DynamoDB
Set	OpenSearch memberikan hasil yang berbeda untuk set DynamoDB tergantun g pada apa yang ada di set. Ketika satu set berisi semua jenis jenis skalar yang sama (misalnya, satu set semua string), kemudian OpenSearch menelan set sebagai array dari jenis itu. Ini berfungsi untuk tipe string, number, Boolean, dan null. Pembatasan untuk masing-masing jenis ini sama dengan batasan untuk skalar jenis itu. Anda juga dapat menyediakan pemetaan untuk set peta dengan menggunakan pemetaan yang sama seperti yang akan Anda gunakan untuk peta. Anda tidak dapat menyediakan satu set tipe campuran.	DynamoDB mendukung jenis yang mewakili set.

Kami menyarankan Anda mengonfigurasi antrian huruf mati (DLQ) di pipeline Ingestion Anda. OpenSearch Jika Anda telah mengonfigurasi antrian, OpenSearch Layanan mengirimkan semua dokumen gagal yang tidak dapat dicerna karena kegagalan pemetaan dinamis ke antrian.

Jika pemetaan otomatis gagal, Anda dapat menggunakan `template_type` dan `template_content` dalam konfigurasi pipeline untuk menentukan aturan pemetaan eksplisit. Atau, Anda dapat membuat templat pemetaan langsung di domain atau koleksi penelusuran sebelum memulai pipeline.

Batasan

Pertimbangkan batasan berikut saat Anda menyiapkan pipeline OpenSearch Ingestion untuk DynamoDB:

- Integrasi OpenSearch Ingestion dengan DynamoDB saat ini tidak mendukung konsumsi lintas wilayah. Tabel DynamoDB OpenSearch dan pipa Ingestion Anda harus sama. Wilayah AWS
- Tabel DynamoDB OpenSearch dan pipa Ingestion Anda harus sama. Akun AWS
- Pipeline OpenSearch Ingestion hanya mendukung satu tabel DynamoDB sebagai sumbernya.
- DynamoDB Streams hanya menyimpan data dalam log hingga 24 jam. Jika konsumsi dari snapshot awal tabel besar membutuhkan waktu 24 jam atau lebih, akan ada beberapa kehilangan data awal. Untuk mengurangi kehilangan data ini, perkirakan ukuran tabel dan konfigurasikan unit komputasi yang sesuai dari OpenSearch pipa Ingestion.

Menggunakan pipa OpenSearch Ingestion dengan Amazon Managed Streaming for Apache Kafka

Anda dapat menggunakan plugin Kafka untuk menyerap data dari Amazon Managed Streaming for Apache Kafka (Amazon OpenSearch MSK) ke dalam pipeline Ingestion Anda. Dengan Amazon MSK, Anda dapat membangun dan menjalankan aplikasi yang menggunakan Apache Kafka untuk memproses data streaming. OpenSearch Penggunaan konsumsi AWS PrivateLink untuk terhubung ke Amazon MSK.

Topik

Prasyarat

Sebelum Anda membuat pipeline OpenSearch Ingestion, lakukan langkah-langkah berikut:

1. Buat klaster MSK Amazon dengan mengikuti langkah-langkah dalam Membuat klaster di Panduan Pengembang Amazon Managed Streaming for Apache Kafka.
   - Untuk tipe Cluster, pilih Provisioned. OpenSearch Ingestion tidak mendukung kluster MSK Tanpa Server.

2. Setelah cluster memiliki status Aktif, ikuti langkah-langkah di Aktifkan konektivitas multi-VPC.
3. Ikuti langkah-langkah di Lampirkan kebijakan klaster ke klaster MSK untuk melampirkan salah satu kebijakan berikut, tergantung pada apakah klaster dan pipeline Anda berada dalam kondisi yang sama Akun AWS. Kebijakan ini memungkinkan OpenSearch Ingestion untuk membuat AWS PrivateLink sambungan ke klaster MSK Amazon Anda dan membaca data dari topik Kafka. Pastikan Anda memperbarui `resource` dengan ARN Anda sendiri.

   Kebijakan berikut berlaku jika klaster dan pipeline Anda berada dalam kondisi yang sama Akun AWS:

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Effect": "Allow",
      "Principal": {
        "Service": "osis.amazonaws.com"
      },
      "Action": [
        "kafka:CreateVpcConnection",
        "kafka:DescribeCluster",
        "kafka:DescribeClusterV2"
      ],
      "Resource": "arn:aws:kafka:us-east-1:{account-id}:cluster/cluster-name/cluster-
id"
    },
    {
      "Effect": "Allow",
      "Principal": {
        "Service": "osis-pipelines.amazonaws.com"
      },
      "Action": [
        "kafka:CreateVpcConnection",
        "kafka:GetBootstrapBrokers",
        "kafka:DescribeCluster",
        "kafka:DescribeClusterV2"
      ],
      "Resource": "arn:aws:kafka:us-east-1:{account-id}:cluster/cluster-name/cluster-
id"
    }
  ]
```

```
}
```

Jika klaster MSK Anda berbeda Akun AWS dari pipeline Anda, lampirkan kebijakan berikut sebagai gantinya. ARN untuk ARN AWS `principal` harus ARN untuk peran pipeline yang sama yang Anda berikan ke konfigurasi YAMM pipleine Anda:

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Effect": "Allow",
      "Principal": {
        "Service": "osis.amazonaws.com"
      },
      "Action": [
        "kafka:CreateVpcConnection",
        "kafka:DescribeCluster",
        "kafka:DescribeClusterV2"
      ],
      "Resource": "arn:aws:kafka:us-east-1:{msk-account-id}:cluster/cluster-
name/cluster-id"
    },
    {
      "Effect": "Allow",
      "Principal": {
        "Service": "osis-pipelines.amazonaws.com"
      },
      "Action": [
        "kafka:CreateVpcConnection",
        "kafka:GetBootstrapBrokers",
        "kafka:DescribeCluster",
        "kafka:DescribeClusterV2"
      ],
      "Resource": "arn:aws:kafka:us-east-1:{msk-account-id}:cluster/cluster-
name/cluster-id"
    },
    {
      "Effect": "Allow",
      "Principal": {
        "AWS": "arn:aws:iam::{pipeline-account-id}:role/pipeline-role"
      },
      "Action": [
        "kafka-cluster:*",
```

```
        "kafka:*"
      ],
      "Resource": [
        "arn:aws:kafka:us-east-1:{msk-account-id}:cluster/cluster-name/cluster-id",
        "arn:aws:kafka:us-east-1:{msk-account-id}:topic/cluster-name/cluster-id/*",
        "arn:aws:kafka:us-east-1:{msk-account-id}:group/cluster-name/*"
      ]
    }
  ]
}
```

4. Buat topik Kafka dengan mengikuti langkah-langkah di Buat topik. Pastikan itu *BootstrapServerString* adalah salah satu URL bootstrap titik akhir pribadi (Single-vPC). Nilai untuk `--replication-factor` harus 2 atau3, berdasarkan jumlah zona yang dimiliki kluster MSK Anda. Nilai untuk setidaknya `--partitions` harus`10`.
5. Menghasilkan dan mengkonsumsi data dengan mengikuti langkah-langkah dalam Menghasilkan dan mengkonsumsi data. Sekali lagi, pastikan itu *BootstrapServerString* adalah salah satu URL bootstrap titik akhir pribadi (Single-vPC) Anda.

Langkah 1: Konfigurasikan peran pipeline

Setelah klaster MSK Anda disiapkan, tambahkan izin Kafka berikut dalam peran pipeline yang ingin Anda gunakan dalam konfigurasi pipeline Anda:

```
{
    "Version": "2012-10-17",
    "Statement": [
        {
            "Effect": "Allow",
            "Action": [
                "kafka-cluster:Connect",
                "kafka-cluster:AlterCluster",
                "kafka-cluster:DescribeCluster",
                "kafka:DescribeClusterV2",
                "kafka:GetBootstrapBrokers"
            ],
            "Resource": [
                "arn:aws:kafka:us-east-1:{account-id}:cluster/cluster-name/cluster-id"
            ]
        },
        {
```

```
            "Effect": "Allow",
            "Action": [
                "kafka-cluster:*Topic*",
                "kafka-cluster:ReadData"
            ],
            "Resource": [
                "arn:aws:kafka:us-east-1:{account-id}:topic/cluster-name/cluster-
id/topic-name"
            ]
        },
        {
            "Effect": "Allow",
            "Action": [
                "kafka-cluster:AlterGroup",
                "kafka-cluster:DescribeGroup"
            ],
            "Resource": [
                "arn:aws:kafka:us-east-1:{account-id}:group/cluster-name/*"
            ]
        }
    ]
}
```

Langkah 2: Buat pipa

Anda kemudian dapat mengonfigurasi pipeline OpenSearch Ingestion seperti berikut ini, yang menentukan Kafka sebagai sumbernya:

```
version: "2"
log-pipeline:
  source:
    kafka:
      acknowledgements: true
      topics:
      - name: "topic-name"
        group_id: "group-id"
        serde_format: "json"/"plaintext"
      aws:
        msk:
          arn: "arn:aws:iam::{account-id}:role/cluster-role"
        region: "us-west-2"
        sts_role_arn: "arn:aws:iam::{account-id}:role/pipeline-role"
      schema:                                # Optional
```

```
      type: "aws_glue"
  processor:
  - grok:
      match:
        log:
        - "%{COMMONAPACHELOG}"
  - date:
      destination: "@timestamp"
      from_time_received: true
  sink:
  - opensearch:
      hosts: ["https://search-domain-endpoint.us-east-1.es.amazonaws.com"]
      index: "index_name"
      aws_sts_role_arn: "arn:aws:iam::{account-id}:role/pipeline-role"
      aws_region: "us-east-1"
      aws_sigv4: true
```

Anda dapat menggunakan cetak biru AWS-MSKPipeline untuk membuat pipeline ini. Untuk informasi selengkapnya, lihat the section called “Menggunakan cetak biru untuk membuat pipeline”.

Langkah 3: (Opsional) Gunakan Registri AWS Glue Skema

Saat Anda menggunakan OpenSearch Ingestion dengan Amazon MSK, Anda dapat menggunakan format data AVRO untuk skema yang dihosting di Registri Skema. AWS Glue Dengan AWS Glue Schema Registry, Anda dapat menemukan, mengontrol, dan mengembangkan skema aliran data secara terpusat.

Untuk menggunakan opsi ini, aktifkan skema `type` dalam konfigurasi pipeline Anda:

```
schema:
  type: "aws_glue"
```

Anda juga harus AWS Glue memberikan izin akses baca dalam peran pipeline Anda. Anda dapat menggunakan kebijakan AWS terkelola yang disebut AWSGlueSchemaRegistryReadonlyAccess. Selain itu, registri Anda harus sama Akun AWS dan Region sebagai pipeline OpenSearch Ingestion Anda.

Langkah 4: (Opsional) Konfigurasikan unit komputasi yang direkomendasikan (OCU) untuk pipeline MSK Amazon

Setiap unit komputasi memiliki satu konsumen per topik. Pialang menyeimbangkan partisi di antara konsumen ini untuk topik tertentu. Namun, ketika jumlah partisi lebih besar dari jumlah konsumen, Amazon MSK menampung beberapa partisi pada setiap konsumen. OpenSearch Ingestion memiliki penskalaan otomatis bawaan untuk meningkatkan atau menurunkan berdasarkan penggunaan CPU atau jumlah catatan yang tertunda dalam pipeline.

Untuk kinerja optimal, distribusikan partisi Anda di banyak unit komputasi untuk pemrosesan paralel. Jika topik memiliki sejumlah besar partisi (misalnya, lebih dari 96, yang merupakan OCU maksimum per pipeline), kami sarankan Anda mengonfigurasi pipeline dengan 1-96 OCU. Ini karena secara otomatis akan menskalakan sesuai kebutuhan. Jika suatu topik memiliki jumlah partisi yang rendah (misalnya, kurang dari 96), pertahankan unit komputasi maksimum sama dengan jumlah partisi.

Jika pipeline memiliki lebih dari satu topik, pilih topik dengan jumlah partisi tertinggi sebagai referensi untuk mengonfigurasi unit komputasi maksimum. Dengan menambahkan pipeline lain dengan kumpulan OCU baru ke topik dan grup konsumen yang sama, Anda dapat menskalakan throughput hampir secara linier.

Menggunakan pipeline OpenSearch Ingestion dengan Amazon S3

Dengan OpenSearch Ingestion, Anda dapat menggunakan Amazon S3 sebagai sumber atau sebagai tujuan. Saat Anda menggunakan Amazon S3 sebagai sumber, Anda mengirim data ke pipeline OpenSearch Ingestion. Saat Anda menggunakan Amazon S3 sebagai tujuan, Anda menulis data dari pipeline OpenSearch Ingestion ke satu atau beberapa bucket S3.

Topik

Amazon S3 sebagai sumber

Ada dua cara Anda dapat menggunakan Amazon S3 sebagai sumber untuk memproses data—dengan pemrosesan S3-SQS dan dengan pemindaian terjadwal.

Gunakan pemrosesan S3-SQS saat Anda memerlukan pemindaian file yang hampir real-time setelah ditulis ke S3. Anda dapat mengonfigurasi bucket Amazon S3 untuk memunculkan peristiwa kapan saja objek disimpan atau dimodifikasi di dalam bucket. Gunakan pemindaian terjadwal satu kali atau berulang untuk mengumpulkan data proses dalam bucket S3.

Topik

Prasyarat

Untuk menggunakan Amazon S3 sebagai sumber pipeline OpenSearch Ingestion untuk pemindaian terjadwal atau pemrosesan S3-SQS, buat bucket S3 terlebih dahulu.

> **Note**
>
> Jika bucket S3 yang digunakan sebagai sumber dalam pipeline OpenSearch Ingestion berbeda Akun AWS, Anda juga perlu mengaktifkan izin baca lintas akun di bucket. Hal ini memungkinkan pipeline untuk membaca dan memproses data. Untuk mengaktifkan izin lintas akun, lihat Pemilik Bucket yang memberikan izin bucket lintas akun di Panduan Pengguna Amazon S3.
> Jika bucket S3 Anda ada di beberapa akun, gunakan peta`bucket_owners`. Sebagai contoh, lihat Akses S3 lintas akun dalam dokumentasi. OpenSearch

Untuk mengatur pemrosesan S3-SQS, Anda juga perlu melakukan langkah-langkah berikut:

1. Buat antrian Amazon SQS.
2. Aktifkan pemberitahuan acara pada bucket S3 dengan antrean SQS sebagai tujuan.

Langkah 1: Konfigurasikan peran pipeline

Tidak seperti plugin sumber lain yang mendorong data ke pipeline, plugin sumber S3 memiliki arsitektur berbasis baca di mana pipeline menarik data dari sumbernya.

Oleh karena itu, agar pipeline dapat dibaca dari S3, Anda harus menentukan peran dalam konfigurasi sumber S3 pipeline yang memiliki akses ke bucket S3 dan antrean Amazon SQS. Pipeline akan mengambil peran ini untuk membaca data dari antrian.

> **Note**
>
> Peran yang Anda tentukan dalam konfigurasi sumber S3 haruslah peran pipeline. Oleh karena itu, peran pipeline Anda harus berisi dua kebijakan izin terpisah—satu untuk menulis ke wastafel, dan satu lagi untuk menarik dari sumber S3. Anda harus menggunakan yang sama `sts_role_arn` di semua komponen pipa.

Kebijakan contoh berikut menunjukkan izin yang diperlukan untuk menggunakan S3 sebagai sumber:

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Effect": "Allow",
      "Action":[
          "s3:ListBucket",
          "s3:GetBucketLocation",
          "s3:GetObject"
       ],
      "Resource": "arn:aws:s3:::my-bucket/*"
    },
    {
       "Effect":"Allow",
       "Action":"s3:ListAllMyBuckets",
       "Resource":"arn:aws:s3:::*"
    },
    {
      "Effect": "Allow",
      "Action": [
        "sqs:DeleteMessage",
        "sqs:ReceiveMessage",
        "sqs:ChangeMessageVisibility"
      ],
      "Resource": "arn:aws:sqs:us-west-2:{account-id}:MyS3EventSqsQueue"
    }
  ]
```

```
}
```

Anda harus melampirkan izin ini ke peran IAM yang Anda tentukan dalam `sts_role_arn` opsi dalam konfigurasi plugin sumber S3:

```
version: "2"
source:
  s3:
    ...
    aws:
      ...
      sts_role_arn: arn:aws:iam::{account-id}:role/pipeline-role
processor:
  ...
sink:
  - opensearch:
      ...
```

Langkah 2: Buat pipa

Setelah menyiapkan izin, Anda dapat mengonfigurasi pipeline OpenSearch Ingestion tergantung pada kasus penggunaan Amazon S3.

Pemrosesan S3-SQS

Untuk menyiapkan pemrosesan S3-SQS, konfigurasikan pipeline Anda untuk menentukan S3 sebagai sumber dan siapkan notifikasi Amazon SQS:

```
version: "2"
s3-pipeline:
  source:
    s3:
      notification_type: "sqs"
      codec:
        newline: null
      sqs:
        queue_url: "https://sqs.us-east-1.amazonaws.com/{account-id}/ingestion-queue"
      compression: "none"
      aws:
        region: "us-east-1"
        # IAM role that the pipeline assumes to read data from the queue. This role
 must be the same as the pipeline role.
```

```
      sts_role_arn: "arn:aws:iam::{account-id}:role/pipeline-role"
processor:
- grok:
    match:
      log:
      - "%{COMMONAPACHELOG}"
- date:
    destination: "@timestamp"
    from_time_received: true
sink:
- opensearch:
    hosts: ["https://search-domain-endpoint.us-east-1.es.amazonaws.com"]
    index: "index-name"
    aws:
      # IAM role that the pipeline assumes to access the domain sink
      sts_role_arn: "arn:aws:iam::{account-id}:role/pipeline-role"
      region: "us-east-1"
```

Jika Anda mengamati pemanfaatan CPU yang rendah saat memproses file kecil di Amazon S3, pertimbangkan untuk meningkatkan throughput dengan memodifikasi nilai opsi. `workers` Untuk informasi selengkapnya, lihat opsi konfigurasi plugin S3.

Pemindaian terjadwal

Untuk menyiapkan pemindaian terjadwal, konfigurasikan pipeline Anda dengan jadwal pada tingkat pemindaian yang berlaku untuk semua bucket S3 Anda, atau di tingkat bucket. Jadwal tingkat ember atau konfigurasi interval pemindaian selalu menimpa konfigurasi tingkat pemindaian.

Anda dapat mengonfigurasi pemindaian terjadwal dengan pemindaian satu kali, yang ideal untuk migrasi data, atau pemindaian berulang, yang ideal untuk pemrosesan batch.

Untuk mengonfigurasi pipeline agar dapat dibaca dari Amazon S3, gunakan cetak biru Amazon S3 bernama AWS-S3 atau AWS-S3. ScanPipeline ScanSchedulePipeline Anda dapat mengedit `scan` bagian konfigurasi pipeline untuk memenuhi kebutuhan penjadwalan Anda. Untuk informasi selengkapnya, lihat the section called “Menggunakan cetak biru untuk membuat pipeline”.

Pemindaian satu kali

Pemindaian terjadwal satu kali berjalan sekali. Dalam konfigurasi YAMM Anda, Anda dapat menggunakan `start_time` dan `end_time` untuk menentukan kapan Anda ingin objek di bucket dipindai. Atau, Anda dapat menggunakan `range` untuk menentukan interval waktu relatif terhadap waktu saat ini yang Anda inginkan objek dalam ember untuk dipindai.

Misalnya, rentang yang diatur untuk PT4H memindai semua file yang dibuat dalam empat jam terakhir. Untuk mengonfigurasi pemindaian satu kali untuk menjalankan kedua kalinya, Anda harus menghentikan dan memulai ulang pipa. Jika Anda tidak memiliki rentang yang dikonfigurasi, Anda juga harus memperbarui waktu mulai dan berakhir.

Konfigurasi berikut menyiapkan pemindaian satu kali untuk semua bucket dan semua objek di bucket tersebut:

```
version: "2"
log-pipeline:
  source:
    s3:
      codec:
        csv:
      compression: "none"
      aws:
        region: "us-east-1"
        sts_role_arn: "arn:aws:iam::{account-id}:role/pipeline-role"
      acknowledgments: true
      scan:
        buckets:
          - bucket:
              name: my-bucket-1
              filter:
                include_prefix:
                  - Objects1/
                exclude_suffix:
                  - .jpeg
                  - .png
          - bucket:
              name: my-bucket-2
              key_prefix:
                include:
                  - Objects2/
                exclude_suffix:
                  - .jpeg
                  - .png
      delete_s3_objects_on_read: false
  processor:
    - date:
        destination: "@timestamp"
        from_time_received: true
  sink:
```

```
    - opensearch:
        hosts: ["https://search-domain-endpoint.us-east-1.es.amazonaws.com"]
        index: "index-name"
        aws:
          sts_role_arn: "arn:aws:iam::{account-id}:role/pipeline-role"
          region: "us-east-1"
        dlq:
          s3:
            bucket: "my-bucket-1"
            region: "us-east-1"
            sts_role_arn: "arn:aws:iam::{account-id}:role/pipeline-role"
```

Konfigurasi berikut mengatur pemindaian satu kali untuk semua bucket selama jendela waktu yang ditentukan. Ini berarti bahwa S3 hanya memproses objek-objek dengan waktu pembuatan yang termasuk dalam jendela ini.

```
scan:
  start_time: 2023-01-21T18:00:00.000Z
  end_time: 2023-04-21T18:00:00.000Z
  buckets:
    - bucket:
        name: my-bucket-1
        filter:
          include:
            - Objects1/
          exclude_suffix:
            - .jpeg
            - .png
    - bucket:
        name: my-bucket-2
        filter:
          include:
            - Objects2/
          exclude_suffix:
            - .jpeg
            - .png
```

Konfigurasi berikut mengatur pemindaian satu kali pada tingkat pemindaian dan tingkat bucket. Waktu mulai dan berakhir pada tingkat bucket menimpa waktu mulai dan berakhir pada tingkat pemindaian.

```
scan:
```

```
  start_time: 2023-01-21T18:00:00.000Z
  end_time: 2023-04-21T18:00:00.000Z
  buckets:
    - bucket:
        start_time: 2023-01-21T18:00:00.000Z
        end_time: 2023-04-21T18:00:00.000Z
        name: my-bucket-1
        filter:
          include:
            - Objects1/
          exclude_suffix:
            - .jpeg
            - .png
    - bucket:
        start_time: 2023-01-21T18:00:00.000Z
        end_time: 2023-04-21T18:00:00.000Z
        name: my-bucket-2
        filter:
          include:
            - Objects2/
          exclude_suffix:
            - .jpeg
            - .png
```

Menghentikan pipa menghilangkan referensi yang sudah ada sebelumnya tentang objek apa yang telah dipindai oleh pipa sebelum berhenti. Jika pipa pemindaian tunggal dihentikan, itu akan memindai ulang semua objek lagi setelah dimulai, bahkan jika mereka sudah dipindai. Jika Anda perlu menghentikan pipa pemindaian tunggal, disarankan Anda mengubah jendela waktu Anda sebelum memulai pipa lagi.

Jika Anda perlu memfilter objek berdasarkan waktu mulai dan waktu akhir, menghentikan dan memulai pipeline Anda adalah satu-satunya pilihan. Jika Anda tidak perlu memfilter berdasarkan waktu mulai dan waktu akhir, Anda dapat memfilter objek berdasarkan nama. Flitering dengan nama tidak mengharuskan Anda untuk berhenti dan memulai pipeline Anda. Untuk melakukan ini, gunakan `include_prefix` dan`exclude_suffix`.

Pemindaian berulang

Pemindaian terjadwal berulang menjalankan pemindaian bucket S3 yang Anda tentukan secara berkala dan terjadwal. Anda hanya dapat mengonfigurasi interval ini pada tingkat pemindaian karena konfigurasi tingkat bucket individual tidak didukung.

Dalam konfigurasi YAMM Anda, `interval` menentukan frekuensi pemindaian berulang, dan bisa antara 30 detik dan 365 hari. Yang pertama dari pemindaian ini selalu terjadi ketika Anda membuat pipa. `count`Mendefinisikan jumlah total instance pemindaian.

Konfigurasi berikut mengatur pemindaian berulang, dengan penundaan 12 jam antara pemindaian:

```
scan:
  scheduling:
    interval: PT12H
    count: 4
  buckets:
    - bucket:
        name: my-bucket-1
        filter:
          include:
            - Objects1/
          exclude_suffix:
            - .jpeg
            - .png
    - bucket:
        name: my-bucket-2
        filter:
          include:
            - Objects2/
          exclude_suffix:
            - .jpeg
            - .png
```

Amazon S3 sebagai tujuan

Untuk menulis data dari pipeline OpenSearch Ingestion ke bucket S3, gunakan cetak biru bernama AWS-S3 untuk membuat pipeline dengan sink S3 SinkLogPipeline. Pipeline ini merutekan data selektif ke OpenSearch sink dan secara bersamaan mengirimkan semua data untuk arsip di S3. Untuk informasi selengkapnya, lihat the section called “Menggunakan cetak biru untuk membuat pipeline”.

Saat Anda membuat wastafel S3, Anda dapat menentukan pemformatan pilihan Anda dari berbagai codec wastafel. Misalnya, jika Anda ingin menulis data dalam format kolom, pilih codec Parket atau Avro. Jika Anda lebih suka format berbasis baris, pilih JSON atau ND-JSON. Untuk menulis data ke S3 dalam skema tertentu, Anda juga dapat menentukan skema inline dalam codec sink menggunakan format Avro.

Contoh berikut mendefinisikan skema inline di wastafel S3:

```
- s3:
  codec:
    parquet:
      schema: >
        {
           "type" : "record",
           "namespace" : "org.vpcFlowLog.examples",
           "name" : "VpcFlowLog",
           "fields" : [
             { "name" : "version", "type" : "string"},
             { "name" : "srcport", "type": "int"},
             { "name" : "dstport", "type": "int"},
             { "name" : "start", "type": "int"},
             { "name" : "end", "type": "int"},
             { "name" : "protocol", "type": "int"},
             { "name" : "packets", "type": "int"},
             { "name" : "bytes", "type": "int"},
             { "name" : "action", "type": "string"},
             { "name" : "logStatus", "type" : "string"}
           ]
         }
```

Saat Anda menentukan skema ini, tentukan superset dari semua kunci yang mungkin ada di berbagai jenis peristiwa yang dikirim pipeline Anda ke wastafel.

Misalnya, jika suatu peristiwa memiliki kemungkinan kunci hilang, tambahkan kunci itu dalam skema Anda dengan `null` nilai. Deklarasi nilai nol memungkinkan skema untuk memproses data yang tidak seragam (di mana beberapa peristiwa memiliki kunci ini dan yang lainnya tidak). Ketika peristiwa yang masuk memang memiliki kunci-kunci ini, nilainya ditulis ke sink.

Definisi skema ini bertindak sebagai filter yang hanya memungkinkan kunci yang ditentukan dikirim ke sink, dan menjatuhkan kunci yang tidak ditentukan dari peristiwa yang masuk.

Anda juga dapat menggunakan `include_keys` dan `exclude_keys` di wastafel Anda untuk memfilter data yang diarahkan ke sink lain. Kedua filter ini saling eksklusif, jadi Anda hanya dapat menggunakan satu per satu dalam skema Anda. Selain itu, Anda tidak dapat menggunakannya dalam skema yang ditentukan pengguna.

Untuk membuat jaringan pipa dengan filter seperti itu, gunakan AWSSinkFilterWithSchemaPipelinecetak biru. Untuk informasi selengkapnya, lihat the section called "Menggunakan cetak biru untuk membuat pipeline".

Akun lintas Amazon S3 sebagai sumber

Anda dapat memberikan akses di seluruh akun dengan Amazon S3 sehingga pipeline OpenSearch Ingestion dapat mengakses bucket S3 di akun lain sebagai sumber. Konfigurasi YAMM berikut memungkinkan akses di seluruh akun ke bucket Amazon S3 sebagai sumber:

```
s3-pipeline:
 source:
  s3:
   notification_type: "sqs"
   codec:
    csv:
     delimiter: ","
     quote_character: "\""
     detect_header: True
   sqs:
    queue_url: "https://sqs.ap-northeast-1.amazonaws.com/401447383613/test-s3-queue"
   bucket_owners:
    user-role-1234567890: 1234567890 # User1
    user-role-12345678891: 1234567891 # User2
   compression: "gzip"
```

Menggunakan pipa OpenSearch Ingestion dengan Amazon Security Lake

Anda dapat menggunakan plugin sumber S3 untuk menyerap data dari Amazon Security Lake ke dalam pipeline OpenSearch Ingestion Anda. Security Lake secara otomatis memusatkan data keamanan dari AWS lingkungan, lingkungan lokal, dan penyedia SaaS ke dalam data lake yang dibuat khusus. Anda dapat membuat langganan yang mereplikasi data dari Security Lake ke pipeline OpenSearch Ingestion Anda, yang kemudian menuliskannya ke domain OpenSearch Layanan atau OpenSearch koleksi Tanpa Server Anda.

Untuk mengonfigurasi pipeline agar dibaca dari Security Lake, gunakan cetak biru Security Lake bernama AWS - S3ParqueToCSFPipeline. SecurityLake Cetak biru mencakup konfigurasi default untuk menelan file parket Open Cybersecurity Schema Framework (OCSF) dari Security Lake. Untuk informasi selengkapnya, lihat the section called "Menggunakan cetak biru untuk membuat pipeline".

Topik

Prasyarat

Sebelum Anda membuat pipeline OpenSearch Ingestion, lakukan langkah-langkah berikut:

- Aktifkan Danau Keamanan.
- Buat pelanggan di Security Lake.
  - Pilih sumber yang ingin Anda konsumsi ke dalam pipa Anda.
  - Untuk kredensyal Pelanggan, tambahkan ID Akun AWS tempat Anda ingin membuat pipeline. Untuk ID eksternal, tentukan`OpenSearchIngestion-{accountid}`.
  - Untuk metode akses Data, pilih S3.
  - Untuk detail Pemberitahuan, pilih antrian SQS.

Saat Anda membuat pelanggan, Security Lake secara otomatis membuat dua kebijakan izin sebaris—satu untuk S3 dan satu untuk SQS. Kebijakan mengambil format berikut: `AmazonSecurityLake-{12345}-S3` dan`AmazonSecurityLake-{12345}-SQS`. Untuk memungkinkan pipeline mengakses sumber pelanggan, Anda harus mengaitkan izin yang diperlukan dengan peran pipeline Anda.

Langkah 1: Konfigurasikan peran pipeline

Buat kebijakan izin baru di IAM yang hanya menggabungkan izin yang diperlukan dari dua kebijakan yang dibuat secara otomatis Security Lake. Contoh kebijakan berikut menunjukkan hak istimewa paling sedikit yang diperlukan untuk pipeline OpenSearch Ingestion untuk membaca data dari beberapa sumber Security Lake:

```
{
   "Version":"2012-10-17",
   "Statement":[
      {
         "Effect":"Allow",
         "Action":[
            "s3:GetObject"
         ],
```

```
         "Resource":[
            "arn:aws:s3:::aws-security-data-lake-{region}-abcde/aws/
LAMBDA_EXECUTION/1.0/*",
            "arn:aws:s3:::aws-security-data-lake-{region}-abcde/aws/S3_DATA/1.0/*",
            "arn:aws:s3:::aws-security-data-lake-{region}-abcde/aws/VPC_FLOW/1.0/*",
            "arn:aws:s3:::aws-security-data-lake-{region}-abcde/aws/ROUTE53/1.0/*",
            "arn:aws:s3:::aws-security-data-lake-{region}-abcde/aws/SH_FINDINGS/1.0/*"
         ]
      },
      {
         "Effect":"Allow",
         "Action":[
            "sqs:ReceiveMessage",
            "sqs:DeleteMessage"
         ],
         "Resource":[
            "arn:aws:sqs:{region}:{account-id}:AmazonSecurityLake-abcde-Main-Queue"
         ]
      }
   ]
}
```

Important

Security Lake tidak mengelola kebijakan peran pipeline untuk Anda. Jika Anda menambah atau menghapus sumber dari langganan Security Lake, Anda harus memperbarui kebijakan secara manual. Security Lake membuat partisi untuk setiap sumber log, jadi Anda perlu menambahkan atau menghapus izin secara manual dalam peran pipeline.

Anda harus melampirkan izin ini ke peran IAM yang Anda tentukan dalam `sts_role_arn` opsi dalam konfigurasi plugin sumber S3, di bawah. `sqs`

```
version: "2"
source:
  s3:
    ...
    sqs:
      queue_url: "https://sqs.{region}.amazonaws.com/{account-id}/
AmazonSecurityLake-abcde-Main-Queue"
    aws:
      ...
```

```
      sts_role_arn: arn:aws:iam::{account-id}:role/pipeline-role
processor:
  ...
sink:
  - opensearch:
      ...
```

Langkah 2: Buat pipa

Setelah Anda menambahkan izin ke peran pipeline, gunakan cetak biru AWS- SecurityLake S3ParqueToCSFPipeline untuk membuat pipeline. Untuk informasi selengkapnya, lihat the section called “Menggunakan cetak biru untuk membuat pipeline”.

Anda harus menentukan `queue_url` opsi dalam konfigurasi `s3` sumber, yang merupakan URL antrian Amazon SQS untuk dibaca. Untuk memformat URL, cari titik akhir Langganan dalam konfigurasi pelanggan dan ubah `arn:aws:` ke. `https://` Misalnya, `https://sqs.{region}.amazonaws.com/{account-id}/AmazonSecurityLake-abdcef-Main-Queue`.

`sts_role_arn`Yang Anda tentukan dalam konfigurasi sumber S3 haruslah ARN dari peran pipeline.

Menggunakan pipa OpenSearch Ingestion dengan Fluent Bit

Contoh file konfigurasi Fluent Bit ini mengirimkan data log dari Fluent Bit ke pipeline OpenSearch Ingestion. Untuk informasi selengkapnya tentang menelan data log, lihat Analisis Log di dokumentasi Penyedia Data.

Perhatikan hal berikut:

- `host`Nilainya harus menjadi titik akhir pipeline Anda. Misalnya, *`pipeline-endpoint`*`.us-east-1.osis.amazonaws.com`.
- Nilai `aws_service` haruslah `osis`.
- `aws_role_arn`Nilainya adalah ARN dari peran AWS IAM untuk diasumsikan dan digunakan klien untuk otentikasi Signature Version 4.

```
[INPUT]
  name                tail
  refresh_interval    5
  path                /var/log/test.log
```

```
  read_from_head        true

[OUTPUT]
  Name http
  Match *
  Host pipeline-endpoint.us-east-1.osis.amazonaws.com
  Port 443
  URI /log/ingest
  Format json
  aws_auth true
  aws_region us-east-1
  aws_service osis
  aws_role_arn arn:aws:iam::{account-id}:role/ingestion-role
  Log_Level trace
  tls On
```

Anda kemudian dapat mengonfigurasi pipeline OpenSearch Ingestion seperti berikut ini, yang memiliki HTTP sebagai sumbernya:

```
version: "2"
unaggregated-log-pipeline:
  source:
    http:
      path: "/log/ingest"
  processor:
    - grok:
        match:
          log:
            - "%{TIMESTAMP_ISO8601:timestamp} %{NOTSPACE:network_node}
 %{NOTSPACE:network_host} %{IPORHOST:source_ip}:%{NUMBER:source_port:int} ->
 %{IPORHOST:destination_ip}:%{NUMBER:destination_port:int} %{GREEDYDATA:details}"
    - grok:
        match:
          details:
            - "'%{NOTSPACE:http_method} %{NOTSPACE:http_uri}' %{NOTSPACE:protocol}"
            - "TLS%{NOTSPACE:tls_version} %{GREEDYDATA:encryption}"
            - "%{NUMBER:status_code:int} %{NUMBER:response_size:int}"
    - delete_entries:
        with_keys: ["details", "log"]

  sink:
    - opensearch:
        hosts: ["https://search-domain-endpoint.us-east-1.es.amazonaws.com"]
```

```
        index: "index_name"
        index_type: custom
        bulk_size: 20
        aws:
          # IAM role that the pipeline assumes to access the domain sink
          sts_role_arn: "arn:aws:iam::{account-id}:role/pipeline-role"
          region: "us-east-1"
```

Menggunakan pipa OpenSearch Ingestion dengan Collector OpenTelemetry

File OpenTelemetry konfigurasi sampel ini mengekspor data jejak dari OpenTelemetry Collector dan mengirimkannya ke pipeline OpenSearch Ingestion. Untuk informasi selengkapnya tentang menelan data jejak, lihat Trace Analytics di dokumentasi Penyedia Data.

Perhatikan hal berikut:

- `endpoint`Nilai harus menyertakan titik akhir pipeline Anda. Misalnya, `https://pipeline-endpoint.us-east-1.osis.amazonaws.com`.
- Nilai `service` haruslah `osis`.
- `compression`Opsi untuk Eksportir OTLP/HTTP harus sesuai dengan `compression` opsi pada sumber pipeline. OpenTelemetry

```
extensions:
  sigv4auth:
    region: "us-east-1"
    service: "osis"

receivers:
  jaeger:
    protocols:
      grpc:

exporters:
  otlphttp:
    traces_endpoint: "https://pipeline-endpoint.us-east-1.osis.amazonaws.com/v1/traces"
    auth:
      authenticator: sigv4auth
    compression: none

service:
```

```
  extensions: [sigv4auth]
  pipelines:
    traces:
      receivers: [jaeger]
      exporters: [otlphttp]
```

Anda kemudian dapat mengonfigurasi pipeline OpenSearch Ingestion seperti berikut ini, yang menentukan plugin jejak OTel sebagai sumbernya:

```
version: "2"
otel-trace-pipeline:
  source:
    otel_trace_source:
      path: "/v1/traces"
  processor:
    - trace_peer_forwarder:
  sink:
    - pipeline:
        name: "trace-pipeline"
    - pipeline:
        name: "service-map-pipeline"
trace-pipeline:
  source:
    pipeline:
      name: "otel-trace-pipeline"
  processor:
    - otel_traces:
  sink:
    - opensearch:
        hosts: ["https://search-domain-endpoint.us-east-1.es.amazonaws.com"]
        index_type: trace-analytics-raw
        aws:
          # IAM role that OpenSearch Ingestion assumes to access the domain sink
          sts_role_arn: "arn:aws:iam::{account-id}:role/pipeline-role"
          region: "us-east-1"

service-map-pipeline:
  source:
    pipeline:
      name: "otel-trace-pipeline"
  processor:
    - service_map:
  sink:
```

```
    - opensearch:
        hosts: ["https://search-domain-endpoint.us-east-1.es.amazonaws.com"]
        index_type: trace-analytics-service-map
        aws:
          # IAM role that the pipeline assumes to access the domain sink
          sts_role_arn: "arn:aws:iam::{account-id}:role/pipeline-role"
          region: "us-east-1"
```

Untuk contoh pipeline lainnya, lihat cetak biru pipeline Trace Analytics. Untuk informasi selengkapnya, lihat the section called “Menggunakan cetak biru untuk membuat pipeline”.

Langkah selanjutnya

Setelah mengekspor data ke pipeline, Anda dapat melakukan kueri dari domain OpenSearch Layanan yang dikonfigurasi sebagai sink untuk pipeline. Sumber daya berikut dapat membantu Anda memulai:

- Hasil pengamatan
- the section called “Trace Analytics”
- the section called “Bahasa Pemrosesan yang Disalurkan”

Migrasi data antara domain dan koleksi menggunakan Amazon Ingestion OpenSearch

Anda dapat menggunakan pipeline OpenSearch Ingestion untuk memigrasikan data antara domain OpenSearch Layanan Amazon atau OpenSearch koleksi VPC Tanpa Server. Untuk melakukannya, Anda menyiapkan pipeline tempat Anda mengonfigurasi satu domain atau koleksi sebagai sumber, dan domain atau koleksi lain sebagai wastafel. Ini secara efektif memigrasikan data Anda dari satu domain atau koleksi ke domain lainnya.

Untuk memigrasikan data, Anda harus memiliki sumber daya berikut:

- Domain OpenSearch layanan sumber atau koleksi OpenSearch VPC Tanpa Server. Domain atau koleksi ini berisi data yang ingin Anda migrasikan. Jika Anda menggunakan domain, domain tersebut harus menjalankan OpenSearch versi 1.0 atau yang lebih baru, atau Elasticsearch versi 7.4 atau yang lebih baru. Domain juga harus memiliki kebijakan akses yang memberikan izin yang sesuai untuk peran pipeline Anda.

- Domain terpisah atau koleksi VPC tempat Anda ingin memigrasikan data Anda. Domain atau koleksi ini akan bertindak sebagai sink pipa.
- Peran pipeline yang akan digunakan OpenSearch Ingestion untuk membaca dan menulis ke koleksi atau domain Anda. Anda menyertakan Nama Sumber Daya Amazon (ARN) peran ini dalam konfigurasi pipeline Anda. Untuk informasi selengkapnya, lihat sumber daya berikut:
  - the section called “Memberikan akses jaringan pipa ke domain”
  - the section called “Memberikan akses jaringan pipa ke koleksi”

Topik

Batasan

Batasan berikut berlaku saat Anda menetapkan domain OpenSearch Layanan atau koleksi OpenSearch Tanpa Server sebagai sink:

- Pipeline tidak dapat menulis ke lebih dari satu domain VPC.
- Anda hanya dapat memigrasikan data ke atau dari koleksi OpenSearch Tanpa Server yang menggunakan akses VPC. Koleksi publik tidak didukung.
- Anda tidak dapat menentukan kombinasi VPC dan domain publik dalam satu konfigurasi pipeline.
- Anda dapat memiliki maksimum 20 sink non-pipa dalam satu konfigurasi pipa.
- Anda dapat menentukan sink dari maksimum tiga yang berbeda Wilayah AWS dalam satu konfigurasi pipa.
- Pipeline dengan beberapa sink mungkin mengalami pengurangan kecepatan pemrosesan dari waktu ke waktu jika salah satu sink mati terlalu lama, atau tidak disediakan dengan kapasitas yang cukup untuk menerima data yang masuk.

OpenSearch Layanan sebagai sumber

Domain atau koleksi yang Anda tentukan sebagai sumber adalah tempat data dimigrasi.

Membuat peran pipeline di IAM

Untuk membuat pipeline OpenSearch Ingestion, Anda harus terlebih dahulu membuat peran pipeline untuk memberikan akses baca dan tulis antara domain atau koleksi. Untuk melakukan ini, lakukan langkah-langkah berikut:

1. Buat kebijakan izin baru di IAM untuk dilampirkan ke peran pipeline. Pastikan Anda mengizinkan izin untuk membaca dari sumber dan menulis ke wastafel. Untuk informasi selengkapnya tentang menyetel izin pipeline IAM untuk domain OpenSearch Layanan, lihat dan. the section called “Memberikan akses jaringan pipa ke domain” the section called “Memberikan akses jaringan pipa ke koleksi”
2. Tentukan izin berikut dalam peran pipeline untuk dibaca dari sumbernya:

```
{
   "Version":"2012-10-17",
   "Statement":[
      {
         "Effect":"Allow",
         "Action":"es:ESHttpGet",
         "Resource":[
            "arn:aws:es:us-east-1:{account-id}:domain/{domain-name}/",
            "arn:aws:es:us-east-1:{account-id}:domain/{domain-name}/_cat/indices",
            "arn:aws:es:us-east-1:{account-id}:domain/{domain-name}/_search",
            "arn:aws:es:us-east-1:{account-id}:domain/{domain-name}/_search/scroll",
            "arn:aws:es:us-east-1:{account-id}:domain/{domain-name}/*/_search"
         ]
      },
      {
         "Effect":"Allow",
         "Action":"es:ESHttpPost",
         "Resource":[
            "arn:aws:es:us-east-1:{account-id}:domain/{domain-name}/*/_search/
point_in_time",
            "arn:aws:es:us-east-1:{account-id}:domain/{domain-name}/*/_search/scroll"
         ]
      },
      {
         "Effect":"Allow",
         "Action":"es:ESHttpDelete",
         "Resource":[
            "arn:aws:es:us-east-1:{account-id}:domain/{domain-name}/_search/
point_in_time",
```

```
            "arn:aws:es:us-east-1:{account-id}:domain/{domain-name}/_search/scroll"
         ]
      }
   ]
}
```

Membuat pipa

Setelah Anda melampirkan kebijakan ke peran pipeline, gunakan cetak biru AWSOpenSearchDataMigrationPipelinemigrasi untuk membuat pipeline. Cetak biru ini mencakup konfigurasi default untuk memigrasikan data antara domain atau koleksi OpenSearch Layanan. Untuk informasi selengkapnya, lihat the section called “Menggunakan cetak biru untuk membuat pipeline”.

> **Note**
>
> OpenSearch Ingestion menggunakan versi dan distribusi domain sumber Anda untuk menentukan mekanisme apa yang akan digunakan untuk migrasi. Beberapa versi mendukung `point_in_time` opsi ini. OpenSearch Tanpa server menggunakan `search_after` opsi karena tidak mendukung `point_in_time` atau. `scroll`

Indeks baru mungkin sedang dalam proses dibuat selama proses migrasi, atau dokumen mungkin diperbarui saat migrasi sedang berlangsung. Karena itu, Anda mungkin perlu melakukan pemindaian tunggal atau beberapa pemindaian data indeks domain Anda untuk mengambil data baru atau yang diperbarui.

Tentukan jumlah pemindaian yang akan dijalankan dengan mengonfigurasi `index_read_count` dan `interval` dalam konfigurasi pipeline. Contoh berikut menunjukkan cara melakukan beberapa pemindaian:

```
scheduling:
    interval: "PT2H"
    index_read_count: 3
    start_time: "2023-06-02T22:01:30.00Z"
```

OpenSearch Ingestion menggunakan konfigurasi berikut untuk memastikan bahwa data Anda ditulis ke indeks yang sama dan mempertahankan ID dokumen yang sama:

```
index: "${getMetadata(\"opensearch-index\")}"
```

```
document_id: "${getMetadata(\"opensearch-document_id\")}"
```

Menentukan beberapa sink domain OpenSearch Layanan

Anda dapat menentukan beberapa domain OpenSearch Layanan publik sebagai tujuan untuk data Anda. Anda dapat menggunakan kemampuan ini untuk melakukan perutean bersyarat atau mereplikasi data yang masuk ke beberapa domain Layanan. OpenSearch Anda dapat menentukan hingga 10 domain OpenSearch Layanan publik yang berbeda sebagai sink.

Dalam contoh berikut, data yang masuk dirutekan secara kondisional ke domain Layanan yang berbeda OpenSearch :

```
...
  route:
    - 2xx_status: "/response >= 200 and /response < 300"
    - 5xx_status: "/response >= 500 and /response < 600"
  sink:
    - opensearch:
        hosts: [ "https://search-response-2xx.us-east-1.es.amazonaws.com" ]
        aws:
          sts_role_arn: "arn:aws:iam::123456789012:role/Example-Role"
          region: "us-east-1"
        index: "response-2xx"
        routes:
          - 2xx_status
    - opensearch:
        hosts: [ "https://search-response-5xx.us-east-1.es.amazonaws.com" ]
        aws:
          sts_role_arn: "arn:aws:iam::123456789012:role/Example-Role"
          region: "us-east-1"
        index: "response-5xx"
        routes:
          - 5xx_status
```

Migrasi data ke koleksi OpenSearch VPC Tanpa Server

Anda dapat menggunakan OpenSearch Ingestion untuk memigrasikan data dari domain OpenSearch Layanan sumber atau koleksi Tanpa OpenSearch Server ke sink koleksi VPC. Anda harus memberikan kebijakan akses jaringan dalam konfigurasi pipeline. Untuk informasi selengkapnya tentang konsumsi data ke dalam koleksi VPC OpenSearch Tanpa Server, lihat. the section called “Tutorial: Menyerap data ke dalam koleksi”

Untuk memigrasikan data ke koleksi VPC

1. Buat koleksi OpenSearch Tanpa Server. Untuk petunjuk, lihat the section called “Tutorial: Menyerap data ke dalam koleksi”.
2. Buat kebijakan jaringan untuk koleksi yang menentukan akses VPC ke titik akhir koleksi dan titik akhir Dasbor. Untuk petunjuk, lihat the section called “Akses jaringan”.
3. Buat peran pipeline jika Anda belum memilikinya. Untuk petunjuk, lihat the section called “Peran pipa”.
4. Buat pipa. Untuk petunjuk, lihat the section called “Menggunakan cetak biru untuk membuat pipeline”.

Menggunakan AWS SDK untuk berinteraksi dengan Amazon OpenSearch Ingestion

Bagian ini mencakup contoh cara menggunakan AWS SDK untuk berinteraksi dengan Amazon OpenSearch Ingestion. Contoh kode menunjukkan cara membuat domain dan pipeline, dan kemudian menelan data ke dalam pipeline.

Topik

- Python

Python

Contoh skrip berikut menggunakan AWS SDK for Python (Boto3)untuk membuat peran pipeline IAM, domain untuk menulis data, dan pipeline untuk menelan data. Kemudian menelan file log sampel ke dalam pipeline menggunakan pustaka `requests` HTTP.

Untuk menginstal dependensi yang diperlukan, jalankan perintah berikut:

```
pip install boto3
pip install botocore
pip install requests
pip install requests-auth-aws-sigv4
```

Di dalam skrip, ganti ID akun dalam kebijakan akses dengan Akun AWS ID Anda. Anda juga dapat memodifikasi file. `region`

```
import boto3
import botocore
from botocore.config import Config
import requests
from requests_auth_aws_sigv4 import AWSSigV4
import time

# Build the client using the default credential configuration.
# You can use the CLI and run 'aws configure' to set access key, secret
# key, and default region.

my_config = Config(
    # Optionally lets you specify a Region other than your default.
    region_name='us-east-1'
)

opensearch = boto3.client('opensearch', config=my_config)
iam = boto3.client('iam', config=my_config)
osis = boto3.client('osis', config=my_config)

domainName = 'test-domain'  # The name of the domain
pipelineName = 'test-pipeline' # The name of the pipeline

def createPipelineRole(iam, domainName):
    """Creates the pipeline role"""
    response = iam.create_policy(
        PolicyName='pipeline-policy',
        PolicyDocument=f'{{\"Version\":\"2012-10-17\",\"Statement\":[{{\"Effect
\":\"Allow\",\"Action\":\"es:DescribeDomain\",\"Resource\":\"arn:aws:es:us-
east-1:123456789012:domain\/{domainName}\"}},{{\"Effect\":\"Allow\",\"Action\":
\"es:ESHttp*\",\"Resource\":\"arn:aws:es:us-east-1:123456789012:domain\/{domainName}\/*
\"}}]}}'
    )
    policyarn = response['Policy']['Arn']

    response = iam.create_role(
        RoleName='PipelineRole',
        AssumeRolePolicyDocument='{\"Version\":\"2012-10-17\",\"Statement\":[{\"Effect
\":\"Allow\",\"Principal\":{\"Service\":\"osis-pipelines.amazonaws.com\"},\"Action\":
\"sts:AssumeRole\"}]}'
    )
    rolename=response['Role']['RoleName']
```

```
    response = iam.attach_role_policy(
        RoleName=rolename,
        PolicyArn=policyarn
    )

    print('Creating pipeline role...')
    time.sleep(10)
    print('Role created: ' + rolename)

def createDomain(opensearch, domainName):
    """Creates a domain to ingest data into"""
    response = opensearch.create_domain(
        DomainName=domainName,
        EngineVersion='OpenSearch_2.3',
        ClusterConfig={
            'InstanceType': 't2.small.search',
            'InstanceCount': 5,
            'DedicatedMasterEnabled': True,
            'DedicatedMasterType': 't2.small.search',
            'DedicatedMasterCount': 3
        },
        # Many instance types require EBS storage.
        EBSOptions={
            'EBSEnabled': True,
            'VolumeType': 'gp2',
            'VolumeSize': 10
        },
        AccessPolicies=f'{{\"Version\":\"2012-10-17\",\"Statement\":[{{\"Effect\":
\"Allow\",\"Principal\":{{\"AWS\":\"arn:aws:iam::123456789012:role\/PipelineRole
\"}},\"Action\":\"es:*\",\"Resource\":\"arn:aws:es:us-east-1:123456789012:domain\/
{domainName}\/*\"}}]}}',
        NodeToNodeEncryptionOptions={
            'Enabled': True
        }
    )
    return(response)

def waitForDomainProcessing(opensearch, domainName):
    """Waits for the domain to be active"""
    try:
        response = opensearch.describe_domain(
            DomainName=domainName
        )
        # Every 30 seconds, check whether the domain is processing.
```

```
        while 'Endpoint' not in response['DomainStatus']:
            print('Creating domain...')
            time.sleep(60)
            response = opensearch.describe_domain(
                DomainName=domainName)

        # Once we exit the loop, the domain is ready for ingestion.
        endpoint = response['DomainStatus']['Endpoint']
        print('Domain endpoint ready to receive data: ' + endpoint)
        createPipeline(osis, endpoint)

    except botocore.exceptions.ClientError as error:
        if error.response['Error']['Code'] == 'ResourceNotFoundException':
            print('Domain not found.')
        else:
            raise error

def createPipeline(osis, endpoint):
    """Creates a pipeline using the domain and pipeline role"""
    try:
        definition = f'version: \"2\"\nlog-pipeline:\n  source:\n    http:\n      path:
 \"/${{pipelineName}}/logs\"\n  processor:\n    - date:\n        from_time_received:
 true\n        destination: \"@timestamp\"\n  sink:\n    - opensearch:\n        hosts:
 [ \"https://{endpoint}\" ]\n        index: \"application_logs\"\n        aws:\n
     sts_role_arn: \"arn:aws:iam::123456789012:role/PipelineRole\"\n          region:
 \"us-east-1\"'
        response = osis.create_pipeline(
            PipelineName=pipelineName,
            MinUnits=4,
            MaxUnits=9,
            PipelineConfigurationBody=definition
        )

        response = osis.get_pipeline(
                PipelineName=pipelineName
        )

        # Every 30 seconds, check whether the pipeline is active.
        while response['Pipeline']['Status'] == 'CREATING':
            print('Creating pipeline...')
            time.sleep(30)
            response = osis.get_pipeline(
                PipelineName=pipelineName)
```

```
        # Once we exit the loop, the pipeline is ready for ingestion.
        ingestionEndpoint = response['Pipeline']['IngestEndpointUrls'][0]
        print('Pipeline ready to ingest data at endpoint: ' + ingestionEndpoint)
        ingestData(ingestionEndpoint)

    except botocore.exceptions.ClientError as error:
        if error.response['Error']['Code'] == 'ResourceAlreadyExistsException':
            print('Pipeline already exists.')
            response = osis.get_pipeline(
                PipelineName=pipelineName
            )
            ingestionEndpoint = response['Pipeline']['IngestEndpointUrls'][0]
            ingestData(ingestionEndpoint)
        else:
            raise error


def ingestData(ingestionEndpoint):
    """Ingests a sample log file into the pipeline"""
    endpoint = 'https://' + ingestionEndpoint
    r = requests.request('POST', f'{endpoint}/log-pipeline/logs',

 data='[{"time":"2014-08-11T11:40:13+00:00","remote_addr":"122.226.223.69","status":"404","requ
 http://www.k2proxy.com//hello.html HTTP/1.1","http_user_agent":"Mozilla/4.0
 (compatible; WOW64; SLCC2;)"}]',
    auth=AWSSigV4('osis'))
    print('Ingesting sample log file into pipeline')
    print('Response: ' + r.text)

def main():
    createPipelineRole(iam, domainName)
    createDomain(opensearch, domainName)
    waitForDomainProcessing(opensearch, domainName)

if __name__ == "__main__":
    main()
```

Kasus penggunaan untuk Amazon OpenSearch Ingestion

Bab ini menunjukkan beberapa kasus penggunaan umum untuk Amazon OpenSearch Ingestion. Daftar ini bukan daftar lengkap. Untuk kemampuan lengkap dari setiap plugin yang didukung, lihat Sumber, Prosesor, dan Tenggelam dalam dokumentasi Data Prepper.

Topik

Pola Grok cocok dengan Amazon Ingestion OpenSearch

Amazon OpenSearch Ingestion menyediakan kemampuan pencocokan pola dengan prosesor Grok. Prosesor Grok didasarkan pada `java-grok` perpustakaan dan mendukung semua pola yang kompatibel. `java-grok`Pustaka dibangun menggunakan pustaka ekspresi `java.util.regex` reguler.

Anda dapat menambahkan pola kustom ke pipeline Anda menggunakan `patterns_definitions` opsi. Saat men-debug pola khusus, Grok Debugger dapat membantu.

Selain contoh-contoh ini, Anda juga dapat menggunakan cetak biru pipa log Apache. Untuk informasi selengkapnya tentang cetak biru, lihat the section called “Menggunakan cetak biru untuk membuat pipeline”.

Topik

Penggunaan dasar

Untuk memulai pencocokan pola, buat pipeline berikut:

```
version: "2"
patten-matching-pipeline:
  source
    ...
  processor:
    - grok:
        match:
          message: ['%{IPORHOST:clientip} \[%{HTTPDATE:timestamp}\]
 %{NUMBER:response_status:int}']
  sink:
    - opensearch:
        # Provide an OpenSearch Service domain endpoint
        # Enable the 'serverless' flag if the sink is an OpenSearch Serverless
 collection
        aws:
          ...
        index: "metrics_for_traces"
        # serverless: true
```

Pesan masuk ke pipeline mungkin memiliki konten berikut:

```
{"message": "127.0.0.1 198.126.12 [10/Oct/2000:13:55:36 -0700] 200"}
```

Pipeline akan menemukan nilai di `message` kunci setiap peristiwa yang masuk dan mencoba mencocokkan polanya. Kata kunci`IPORHOST`,`HTTPDATE`, dan `NUMBER` dibangun ke dalam plugin.

Ketika catatan masuk cocok dengan pola, itu menghasilkan peristiwa internal seperti berikut ini, dengan kunci identifikasi yang diekstrak dari pesan asli.

```
{
  "message":"127.0.0.1 198.126.12 [10/Oct/2000:13:55:36 -0700] 200",
  "response_status":200,
  "clientip":"198.126.12",
  "timestamp":"10/Oct/2000:13:55:36 -0700"
}
```

`match`Konfigurasi untuk prosesor Grok menentukan kunci catatan mana yang cocok dengan pola mana.

Dalam contoh berikut, konfigurasi pencocokan memeriksa log masuk untuk `message` kunci. Jika kunci ada, itu cocok dengan nilai kunci terhadap `SYSLOGBASE` pola, dan kemudian melawan

COMMONAPACHELOG pola. Kemudian memeriksa log untuk timestamp kunci. Jika kunci itu ada, ia mencoba untuk mencocokkan nilai kunci terhadap TIMESTAMP_ISO8601 pola.

```
processor:
  - grok:
      match:
        message: ['%{SYSLOGBASE}', "%{COMMONAPACHELOG}"]
        timestamp: ["%{TIMESTAMP_ISO8601}"]
```

Secara default, plugin berlanjut hingga menemukan kecocokan yang berhasil. Misalnya, jika ada kecocokan yang berhasil terhadap nilai dalam message kunci untuk suatu SYSLOGBASE pola, plugin tidak mencoba untuk mencocokkan pola lainnya. Jika Anda ingin mencocokkan log dengan setiap pola, sertakan break_on_match opsi.

Termasuk tangkapan bernama dan kosong

Sertakan keep_empty_captures opsi dalam konfigurasi pipeline Anda untuk menyertakan tangkapan null, atau named_captures_only opsi untuk hanya menyertakan tangkapan bernama. Tangkapan bernama mengikuti pola%{SYNTAX:SEMANTIC}, sementara tangkapan yang tidak disebutkan namanya mengikuti pola. %{SYNTAX}

Misalnya, Anda dapat memodifikasi konfigurasi Grok di atas untuk menghapus clientip dari %{IPORHOST} pola:

```
processor:
  - grok:
      match:
        message: ['%{IPORHOST} \[%{HTTPDATE:timestamp}\]
 %{NUMBER:response_status:int}']
```

Log grokked yang dihasilkan akan terlihat seperti ini:

```
{
  "message":"127.0.0.1 198.126.12 [10/Oct/2000:13:55:36 -0700] 200",
  "response_status":200,
  "timestamp":"10/Oct/2000:13:55:36 -0700"
}
```

Perhatikan bahwa clientip kunci tidak ada lagi, karena %{IPORHOST} polanya sekarang merupakan tangkapan yang tidak disebutkan namanya.

Namun, jika Anda mengatur `named_captures_only` kefalse:

```
processor:
  - grok:
      match:
        named_captures_only: false
        message: ['%{IPORHOST} \[%{HTTPDATE:timestamp}\] %{NUMBER:message:int}']
```

Log grokked yang dihasilkan akan terlihat seperti ini:

```
{
  "message":"127.0.0.1 198.126.12 [10/Oct/2000:13:55:36 -0700] 200",
  "MONTH":"Oct",
  "YEAR":"2000",
  "response_status":200,
  "HOUR":"13",
  "TIME":"13:55:36",
  "MINUTE":"55",
  "SECOND":"36",
  "IPORHOST":"198.126.12",
  "MONTHDAY":"10",
  "INT":"-0700",
  "timestamp":"10/Oct/2000:13:55:36 -0700"
}
```

Perhatikan bahwa `IPORHOST` tangkapan sekarang muncul sebagai kunci baru, bersama dengan beberapa tangkapan internal yang tidak disebutkan namanya seperti `MONTH` dan. `YEAR` `HTTPDATE`Kata kunci menggunakan pola-pola ini, yang dapat Anda lihat di file pola default.

Tombol menimpa

Sertakan `keys_to_overwrite` opsi untuk menentukan kunci rekaman yang ada untuk ditimpa jika ada tangkapan dengan nilai kunci yang sama.

Misalnya, Anda dapat memodifikasi konfigurasi grok di atas untuk diganti `%{NUMBER:response_status:int}` dengan`%{NUMBER:message:int}`, dan menambahkan `message` ke daftar kunci untuk menimpa.

```
processor:
  - grok:
```

```
      match:
        keys_to_overwrite: ["message"]
        message: ['%{IPORHOST:clientip} \[%{HTTPDATE:timestamp}\]
 %{NUMBER:message:int}']
```

Dalam log grokked yang dihasilkan, pesan asli ditimpa dengan nomor 200.

```
{
  "message":200,
  "clientip":"198.126.12",
  "timestamp":"10/Oct/2000:13:55:36 -0700"
}
```

Menggunakan pola kustom

Sertakan `pattern_definitions` opsi dalam konfigurasi grok Anda untuk menentukan pola kustom.

Konfigurasi berikut membuat pola regex kustom bernama `CUSTOM_PATTERN-1` dan. `CUSTOM_PATTERN-2` Secara default, plugin berlanjut hingga menemukan kecocokan yang berhasil.

```
processor:
  - grok:
      pattern_definitions:
        CUSTOM_PATTERN_1: 'this-is-regex-1'
        CUSTOM_PATTERN_2: '%{CUSTOM_PATTERN_1} REGEX'
      match:
        message: ["%{CUSTOM_PATTERN_2:my_pattern_key}"]
```

Jika Anda menentukan `break_on_match` as`false`, pipeline mencoba mencocokkan semua pola dan mengekstrak kunci dari peristiwa yang masuk:

```
processor:
  - grok:
      pattern_definitions:
        CUSTOM_PATTERN_1: 'this-is-regex-1'
        CUSTOM_PATTERN_2: 'this-is-regex-2'
        CUSTOM_PATTERN_3: 'this-is-regex-3'
        CUSTOM_PATTERN_4: 'this-is-regex-4'
      match:
        message: [ "%{PATTERN1}”, "%{PATTERN2}" ]
```

```
        log: [ "%{PATTERN3}", "%{PATTERN4}" ]
        break_on_match: false
```

Anda dapat menentukan pola kustom Anda sendiri yang akan digunakan untuk pencocokan pola di pipeline. Pada contoh sebelumnya, `my_pattern` akan diekstraksi setelah mencocokkan pola kustom.

Menyimpan tangkapan dengan kunci induk

Sertakan `target_key` opsi dalam konfigurasi grok Anda untuk membungkus semua tangkapan untuk catatan dalam nilai kunci luar tambahan.

Misalnya, Anda dapat memodifikasi konfigurasi grok di atas untuk menambahkan kunci target bernama`grokked`.

```
processor:
   - grok:
       target_key: "grok"
       match:
         message: ['%{IPORHOST} \[%{HTTPDATE:timestamp}\]
 %{NUMBER:response_status:int}']
```

Log grokked yang dihasilkan akan terlihat seperti ini:

```
{
  "message":"127.0.0.1 198.126.12 [10/Oct/2000:13:55:36 -0700] 200",
  "grokked": {
     "response_status":200,
     "clientip":"198.126.12",
     "timestamp":"10/Oct/2000:13:55:36 -0700"
  }
}
```

Pengayaan log dengan Amazon Ingestion OpenSearch

Anda dapat melakukan berbagai jenis pengayaan log dengan Amazon OpenSearch Ingestion. Selain contoh-contoh ini, Anda juga dapat menggunakan cetak biru pipa log generik. Untuk informasi selengkapnya tentang cetak biru, lihat the section called “Menggunakan cetak biru untuk membuat pipeline”.

Topik

Penyaringan

Gunakan prosesor Drop events untuk memfilter peristiwa log tertentu sebelum mengirimnya ke wastafel. Misalnya, Anda mengumpulkan log permintaan web dan hanya ingin menyimpan permintaan yang gagal. Anda membuat pipeline berikut, yang menghapus permintaan apa pun yang responsnya kurang dari 400 sehingga hanya mencatat peristiwa dengan kode status HTTP 400 ke atas yang tersisa.

```
version: "2"
log-pipeline:
  source:
  ...
  processor:
    - grok:
        match:
          log: [ "%{COMMONAPACHELOG_DATATYPED}" ]
    - drop:
        drop_when: "/response < 400"
  sink:
    - opensearch:
        ...
        index: failure_logs
```

`drop_when`Opsi menentukan genap mana yang akan jatuh dari pipa.

Mengekstrak pasangan kunci-nilai dari string

Data log sering menyertakan string pasangan kunci-nilai. Salah satu skenario umum adalah string kueri HTTP. Misalnya, jika pengguna web menanyakan URL pageable, log HTTP mungkin memiliki string kueri HTTP berikut:

```
page=3&q=my-search-term
```

Untuk melakukan analisis menggunakan istilah pencarian, Anda dapat mengekstrak nilai q dari string kueri. Prosesor nilai Kunci memberikan dukungan kuat untuk mengekstraksi kunci dan nilai dari string.

Contoh berikut menggabungkan `split_string` dan `key_value` prosesor untuk mengekstrak parameter kueri dari baris log Apache:

```
version: "2"
pipeline
 ...
  processor:
    - grok:
      match:
        message: [ "%{COMMONAPACHELOG_DATATYPED}" ]
    - split_string:
      entries:
        - source: request
          delimiter: "?"
    - key_value:
      source: "/request/1"
      field_split_characters: "&"
      value_split_characters: "="
      destination: query_params
```

Peristiwa bermutasi

Prosesor peristiwa Mutate yang berbeda memungkinkan Anda mengganti nama, menyalin, menambah, dan menghapus entri acara.

Dalam contoh ini, prosesor pertama menetapkan nilai `debug` kunci `true` jika kunci sudah ada dalam acara tersebut. Prosesor kedua hanya menetapkan `debug` kunci `true` jika kunci tidak ada dalam acara tersebut, karena `overwrite_if_key_exists` diatur ke`true`.

```
...
processor:
  - add_entries:
    entries:
    - key: "debug"
      value: true
```

```
...
processor:
  - add_entries:
    entries:
    - key: "debug"
      value: true
      overwrite_if_key_exists: true
...
```

Anda juga dapat menggunakan string format untuk membangun entri baru dari entri yang ada. Misalnya, `${date}-${time}` akan membuat entri baru berdasarkan nilai-nilai entri yang ada `date` dan`time`.

Misalnya, pipeline berikut menambahkan entri acara baru secara dinamis dari peristiwa yang ada:

```
processor:
  - add_entries:
    entries:
    - key: "key_three"
      format: "${key_one}-${key_two}
```

Misalnya, pertimbangkan acara masuk berikut:

```
{
   "key_one": "value_one",
   "key_two": "value_two"
}
```

Prosesor mengubahnya menjadi acara dengan kunci baru`key_three`, yang menggabungkan nilai-nilai kunci lain dalam acara aslinya.

```
{
   "key_one": "value_one",
   "key_two": "value_two",
   "key_three": "value_one-value_two"
}
```

Senar yang bermutasi

Berbagai prosesor string Mutate menawarkan alat untuk memanipulasi string dalam data yang masuk. Misalnya, jika Anda perlu membagi string menjadi array, gunakan `split_string` prosesor:

```
...
processor:
  - split_string:
    entries:
    - source: "message"
      delimiter: "&"
...
```

Prosesor akan mengubah string seperti a&b&c menjadi `["a", "b", "c"]`.

Mengonversi daftar ke peta

ist-to-mapProsesor L, yang merupakan salah satu prosesor peristiwa Mutate, mengubah daftar objek dalam suatu peristiwa ke peta.

Misalnya, pertimbangkan konfigurasi prosesor berikut:

```
...
processor:
    - list_to_map:
        key: "name"
        source: "A-car-as-list"
        target: "A-car-as-map"
        value_key: "value"
        flatten: true
...
```

Prosesor ini akan mengonversi peristiwa yang berisi daftar objek seperti ini:

```
{
  "A-car-as-list": [
    {
      "name": "make",
      "value": "tesla"
    },
    {
      "name": "model",
      "value": "model 3"
    },
    {
      "name": "color",
      "value": "white"
```

```
    }
  ]
}
```

Ke dalam peta:

```
{
  "A-car-as-map": {
    "make": "tesla",
    "model": "model 3",
    "color": "white"
  }
}
```

Sebagai contoh lain, katakanlah Anda memiliki acara masuk dengan struktur berikut:

```
{
  "mylist" : [
    {
      "somekey" : "a",
      "somevalue" : "val-a1",
      "anothervalue" : "val-a2"
    },
    {
      "somekey" : "b",
      "somevalue" : "val-b1",
      "anothervalue" : "val-b2"
    },
    {
      "somekey" : "b",
      "somevalue" : "val-b3",
      "anothervalue" : "val-b4"
    },
    {
      "somekey" : "c",
      "somevalue" : "val-c1",
      "anothervalue" : "val-c2"
    }
  ]
}
```

Anda dapat menentukan opsi berikut dalam konfigurasi prosesor:

```
...
processor:
    - list_to_map:
        key: "somekey"
        source: "mylist"
        target: "myobject"
        value_key: "value"
        flatten: true
...
```

Prosesor memodifikasi acara dengan menghapus `mylist` dan menambahkan `myobject` objek baru:

```
{
  "myobject" : {
    "a" : [
      {
        "somekey" : "a",
        "somevalue" : "val-a1",
        "anothervalue" : "val-a2"
      }
    ],
    "b" : [
      {
        "somekey" : "b",
        "somevalue" : "val-b1",
        "anothervalue" : "val-b2"
      },
      {
        "somekey" : "b",
        "somevalue" : "val-b3",
        "anothervalue" : "val-b4"
      }
    "c" : [
      {
        "somekey" : "c",
        "somevalue" : "val-c1",
        "anothervalue" : "val-c2"
      }
    ]
  }
}
```

Dalam banyak kasus, Anda mungkin ingin meratakan array untuk setiap kunci. Dalam situasi ini, Anda harus memilih hanya satu objek untuk tetap. Prosesor menawarkan pilihan pertama atau terakhir.

```
...
processor:
    - list_to_map:
        key: "somekey"
        source: "mylist"
        target: "myobject"
        flatten: true
...
```

Struktur acara yang masuk kemudian diratakan sesuai:

```
{
  "myobject" : {
    "a" : {
      "somekey" : "a",
      "somevalue" : "val-a1",
      "anothervalue" : "val-a2"
    },
    "b" : {
      "somekey" : "b",
      "somevalue" : "val-b1",
      "anothervalue" : "val-b2"
    }
    "c" : {
      "somekey" : "c",
      "somevalue" : "val-c1",
      "anothervalue" : "val-c2"
    }
  }
}
```

Anda dapat menggunakan ist-to-map prosesor L untuk memproses AWS WAF log. Misalnya, pertimbangkan contoh log WAF seperti ini:

```
{
    "webaclId": "arn:aws:wafv2:ap-southeast-2:111122223333:regional/webacl/
STMTest/1EXAMPLE-2ARN-3ARN-4ARN-123456EXAMPLE",
    "httpRequest": {
```

```
        "headers": [
            {
                "name": "Host",
                "value": "localhost:1989"
            },
            {
                "name": "User-Agent",
                "value": "curl/7.61.1"
            }
        ]
    }
}
```

Jika pipeline berikut memproses acara:

```
...
processor:
    - list_to_map:
        key: "name"
        source: "httpRequest/headers"
        value_key: "value"
        flatten: true
...
```

Ini akan membuat acara baru berikut:

```
{
    "webaclId": "arn:aws:wafv2:ap-southeast-2:111122223333:regional/webacl/
STMTest/1EXAMPLE-2ARN-3ARN-4ARN-123456EXAMPLE",
    "httpRequest": {
        "headers": [
            {
                "name": "Host",
                "value": "localhost:1989"
            },
            {
                "name": "User-Agent",
                "value": "curl/7.61.1"
            }
        ]
    },
    "Host": "localhost:1989",
    "User-Agent": "curl/7.61.1"
```

```
}
```

Memproses stempel waktu yang masuk

Prosesor Tanggal mem-parsing kunci stempel waktu dari peristiwa yang masuk dengan mengonversinya ke format ISO 8601.

```
...
  processor:
    - date:
        match:
          - key: timestamp
            patterns: ["dd/MMM/yyyy:HH:mm:ss"]
        destination: "@timestamp"
        source_timezone: "America/Los_Angeles"
        destination_timezone: "America/Chicago"
        locale: "en_US"
...
```

Jika pipeline di atas memproses peristiwa berikut:

```
{"timestamp": "10/Feb/2000:13:55:36"}
```

Ini mengubah acara ke dalam format berikut:

```
{
  "timestamp":"10/Feb/2000:13:55:36",
  "@timestamp":"2000-02-10T15:55:36.000-06:00"
}
```

Menghasilkan stempel waktu

Prosesor Tanggal dapat menghasilkan stempel waktu untuk peristiwa yang masuk jika Anda menentukan `@timestamp` opsi tersebut. `destination`

```
...
    processor:
    - date:
        from_time_received: true
        destination: "@timestamp"
...
```

Menurunkan pola tanda baca

Prosesor string Substitute (yang merupakan salah satu prosesor string Mutate) memungkinkan Anda memperoleh pola tanda baca dari peristiwa yang masuk. Dalam contoh pipeline berikut, prosesor akan memindai peristiwa log Apache yang masuk dan mendapatkan pola tanda baca dari mereka.

```
processor:

    - substitute_string:

        entries:

          - source: "message"

            from: "[a-zA-Z0-9_]+"

            to: ""

          - source: "message"

            from: "[ ]+"

            to: "_"
```

Log HTTP Apache yang masuk berikut akan menghasilkan pola tanda baca:

```
[{"message":"10.10.10.11 - admin [19/Feb/2015:15:50:36 -0500] \"GET /big2.pdf
 HTTP/1.1\" 200 33973115 0.202 \"-\" \"Mozilla/5.0 (Macintosh; Intel Mac OS X 10_10_1)
 AppleWebKit/537.36 (KHTML, like Gecko) Chrome/40.0.2214.111 Safari/537.36\""}]

{"message":"..._-_[//:::_-]_\"_/._/.\"_._\"-\"_\"/._(;_)_/._(,_)_/..._/.\""}
```

Anda dapat menghitung pola yang dihasilkan ini dengan melewatkannya melalui prosesor Agregat dengan `count` tindakan.

Agregasi acara dengan Amazon Ingestion OpenSearch

Anda dapat menggunakan Amazon OpenSearch Ingestion untuk mengumpulkan data dari berbagai peristiwa selama periode waktu tertentu. Agregasi peristiwa dapat membantu mengurangi volume log yang tidak perlu dan menangani kasus penggunaan seperti log multi-baris yang masuk sebagai peristiwa terpisah. Prosesor Agregat adalah prosesor stateful yang mengelompokkan peristiwa

berdasarkan nilai untuk satu set kunci identifikasi yang ditentukan, dan melakukan tindakan yang dapat dikonfigurasi pada setiap grup.

Status dalam prosesor Agregat disimpan dalam memori. Misalnya, untuk menggabungkan empat peristiwa menjadi satu, prosesor perlu mempertahankan bagian dari tiga peristiwa pertama. Status kelompok agregat peristiwa disimpan untuk jumlah waktu yang dapat dikonfigurasi. Bergantung pada log Anda, tindakan agregat yang digunakan, dan jumlah opsi memori dalam konfigurasi prosesor, agregasi dapat berlangsung dalam jangka waktu yang lama.

Selain contoh-contoh ini, Anda juga dapat menggunakan agregasi Log dengan cetak biru routing bersyarat. Untuk informasi selengkapnya tentang cetak biru, lihat the section called “Menggunakan cetak biru untuk membuat pipeline”.

Topik

Penggunaan dasar

Contoh pipeline berikut mengekstrak bidang`sourceIp`,`destinationIp`, dan `port` menggunakan prosesor Grok, dan kemudian agregat pada bidang tersebut selama periode 30 detik menggunakan prosesor Agregat dan tindakan. `put_all` Pada akhir 30 detik, log agregat dikirim ke OpenSearch wastafel.

```
version: "2"
aggregate_pipeline:
   source:
     http:
      path: "/${pipelineName}/logs"
   processor:
     - grok:
         match:
           log: ["%{IPORHOST:sourceIp} %{IPORHOST:destinationIp} %{NUMBER:port:int}"]
     - aggregate:
         group_duration: "30s"
         identification_keys: ["sourceIp", "destinationIp", "port"]
         action:
           put_all:
```

```
  sink:
    - opensearch:
        ...
        index: aggregated_logs
```

Misalnya, pertimbangkan kumpulan log berikut:

```
{ "log": "127.0.0.1 192.168.0.1 80", "status": 200 }
{ "log": "127.0.0.1 192.168.0.1 80", "bytes": 1000 }
{ "log": "127.0.0.1 192.168.0.1 80" "http_verb": "GET" }
```

Prosesor Grok akan mengekstrak `identification_keys` untuk membuat log berikut:

```
{ "sourceIp": "127.0.0.1", "destinationIp": "192.168.0.1", "port": 80, "status": 200 }
{ "sourceIp": "127.0.0.1", "destinationIp": "192.168.0.1", "port": 80, "bytes": 1000 }
{ "sourceIp": "127.0.0.1", "destinationIp": "192.168.0.1", "port": 80, "http_verb":
 "GET" }
```

Ketika grup selesai 30 detik setelah log pertama diterima oleh prosesor Agregat, log agregat berikut ditulis ke wastafel:

```
{ "sourceIp": "127.0.0.1", "destinationIp": "192.168.0.1", "port": 80, "status": 200,
 "bytes": 1000, "http_verb": "GET" }
```

Menghapus duplikat

Anda dapat menghapus entri duplikat dengan menurunkan kunci dari peristiwa yang masuk dan menentukan `remove_duplicates` opsi untuk prosesor Agregat. Tindakan ini segera memproses peristiwa pertama untuk grup, dan menghapus semua peristiwa berikut dalam grup itu.

Dalam contoh berikut, acara pertama diproses dengan kunci identifikasi `sourceIp` dan`destinationIp`:

```
{ "sourceIp": "127.0.0.1", "destinationIp": "192.168.0.1", "status": 200 }
```

Pipeline kemudian akan menjatuhkan acara berikut karena memiliki kunci yang sama:

```
{ "sourceIp": "127.0.0.1", "destinationIp": "192.168.0.1", "bytes": 1000 }
```

Pipeline memproses acara ini dan membuat grup baru `sourceIp` karena berbeda:

```
{ "sourceIp": "127.0.0.2", "destinationIp": "192.168.0.1", "bytes": 1000 }
```

Agregasi log dan perutean bersyarat

Anda dapat menggunakan beberapa plugin untuk menggabungkan agregasi log dengan perutean bersyarat. Dalam contoh ini, sub-pipeline `log-aggregate-pipeline` menerima log melalui klien HTTP seperti FluentBit dan mengekstrak nilai-nilai penting dari log dengan mencocokkan nilai dalam `log` kunci terhadap pola log Apache umum.

Dua dari nilai yang diekstrak dari log dengan pola grok meliputi `response` dan`clientip`. Prosesor Aggregate kemudian menggunakan `clientip` nilai, bersama dengan `remove_duplicates` opsi, untuk menjatuhkan setiap log yang berisi `clientip` yang telah diproses dalam yang diberikan`group_duration`.

Tiga rute, atau pernyataan bersyarat, ada dalam pipa. Rute-rute ini memisahkan nilai respons menjadi respons 2xx/3xx, 4xx, dan 5xx. Log dengan status 2xx dan 3xx dikirim ke `aggregated_2xx_3xx` indeks, log dengan status 4xx dikirim ke `aggregated_4xx` indeks, dan log dengan status 5xx dikirim ke indeks. `aggregated_5xx`

```
version: "2"
log-aggregate-pipeline:
  source:
    http:
      # Provide the path for ingestion. ${pipelineName} will be replaced with pipeline
 name configured for this pipeline.
      # In this case it would be "/log-aggregate-pipeline/logs". This will be the
 FluentBit output URI value.
      path: "/${pipelineName}/logs"
  processor:
    - grok:
        match:
          log: [ "%{COMMONAPACHELOG_DATATYPED}" ]
    - aggregate:
        identification_keys: ["clientip"]
        action:
          remove_duplicates:
        group_duration: "180s"
  route:
    - 2xx_status: "/response >= 200 and /response < 300"
    - 3xx_status: "/response >= 300 and /response < 400"
    - 4xx_status: "/response >= 400 and /response < 500"
```

```
    - 5xx_status: "/response >= 500 and /response < 600"
  sink:
    - opensearch:
        ...
        index: "aggregated_2xx_3xx"
        routes:
          - 2xx_status
          - 3xx_status
    - opensearch:
        ...
        index: "aggregated_4xx"
        routes:
          - 4xx_status
    - opensearch:
        ...
        index: "aggregated_5xx"
        routes:
          - 5xx_status
```

Menurunkan metrik dari log dengan Amazon Ingestion OpenSearch

Anda dapat menggunakan Amazon OpenSearch Ingestion untuk mendapatkan metrik dari log. Contoh pipeline berikut menerima log masuk menggunakan plugin sumber HTTP dan prosesor Grok. Kemudian menggunakan prosesor Aggregate untuk mengekstrak metrik yang `bytes` dikumpulkan melalui jendela 30 detik dan menurunkan histogram dari hasilnya.

Pipa keseluruhan berisi dua sub-pipeline:

- `apache-log-pipeline-with-metrics`— Menerima log melalui klien HTTP seperti FluentBit, mengekstrak nilai-nilai penting dari log dengan mencocokkan nilai dalam `log` kunci terhadap pola log Apache umum grok, dan kemudian meneruskan log grokked ke `log-to-metrics-pipeline` sub-pipeline dan ke indeks bernama. OpenSearch `logs`
- `log-to-metrics-pipeline`— Menerima log grokked dari `apache-log-pipeline-with-metrics` sub-pipeline, mengumpulkan log dan memperoleh metrik histogram `bytes` berdasarkan nilai dalam dan kunci. `clientip` `request` Akhirnya, ia mengirimkan metrik histogram ke indeks bernama OpenSearch . `histogram_metrics`

```
version: "2"
apache-log-pipeline-with-metrics:
  source:
```

```
    http:
      # Provide the path for ingestion. ${pipelineName} will be replaced with pipeline
 name configured for this pipeline.
      # In this case it would be "/apache-log-pipeline-with-metrics/logs". This will be
 the FluentBit output URI value.
      path: "/${pipelineName}/logs"
  processor:
    - grok:
        match:
          log: [ "%{COMMONAPACHELOG_DATATYPED}" ]
  sink:
    - opensearch:
        ...
        index: "logs"
    - pipeline:
        name: "log-to-metrics-pipeline"

log-to-metrics-pipeline:
  source:
    pipeline:
      name: "apache-log-pipeline-with-metrics"
  processor:
    - aggregate:
        # Specify the required identification keys
        identification_keys: ["clientip", "request"]
        action:
          histogram:
            # Specify the appropriate values for each of the following fields
            key: "bytes"
            record_minmax: true
            units: "bytes"
            buckets: [0, 25000000, 50000000, 75000000, 100000000]
        # Pick the required aggregation period
        group_duration: "30s"
  sink:
    - opensearch:
        ...
        index: "histogram_metrics"
```

Selain contoh ini, Anda juga dapat menggunakan cetak biru Log to metric pipeline. Untuk informasi selengkapnya tentang cetak biru, lihat the section called “Menggunakan cetak biru untuk membuat pipeline”.

Lacak Analytics dengan Amazon OpenSearch Ingestion

Anda dapat menggunakan Amazon OpenSearch Ingestion untuk mengumpulkan data OpenTelemetry jejak dan mengubahnya untuk digunakan dalam OpenSearch Layanan. Contoh pipeline berikut menggunakan tiga sub-pipeline untuk memantau Trace Analytics:`entry-pipeline`,`span-pipeline`, dan. `service-map-pipeline`

OpenTelemetry sumber jejak

Plugin sumber jejak Otel menerima data jejak dari Kolektor. OpenTelemetry Plugin mengikuti OpenTelemetry Protokol dan secara resmi mendukung enkripsi standar industri HTTPS.

Prosesor

Anda dapat menggunakan prosesor berikut untuk Trace Analytics:

- Tel trace — Menerima koleksi catatan rentang dari sumber dan melakukan pemrosesan stateful, ekstraksi, dan penyelesaian bidang.
- Grup jejak OTel - Mengisi bidang grup jejak yang hilang dalam kumpulan catatan rentang.
- Peta layanan — Melakukan pra-pemrosesan untuk melacak data dan membangun metadata untuk menampilkan dasbor peta layanan.

OpenSearch wastafel

Plugin OpenSearch sink menyediakan indeks dan templat indeks yang khusus untuk Trace Analytics. OpenSearch Indeks berikut khusus untuk Trace Analytics:

- `otel-v1-apm-span`— Menyimpan output dari prosesor jejak OTel.
- `otel-v1-apm-service-map`— Menyimpan output dari prosesor Service-map.

Konfigurasi alur

Contoh pipeline berikut mendukung Observability for OpenSearch Dashboards. Sub-pipeline (`entry-pipeline`) pertama menerima data dari OpenTelemetry Collector dan menggunakan dua sub-pipeline lainnya sebagai sink.

`span-pipeline`Sub-pipeline mem-parsing data jejak dan memperkaya dan menyerap dokumen rentang ke dalam indeks rentang. Agregat `service-map-pipeline` sub-pipeline melacak ke peta layanan dan menulis dokumen ke indeks peta layanan.

```
version: "2"
entry-pipeline:
  source:
    otel_trace_source:
      # Provide the path for ingestion. This will be the endpoint URI path in the
 OpenTelemetry Exporter configuration.
      # ${pipelineName} will be replaced with the sub-pipeline name. In this case it
 would be "/entry-pipeline/v1/traces".
      path: "/${pipelineName}/v1/traces"
  processor:
    - trace_peer_forwarder
  sink:
    - pipeline:
        name: "span-pipeline"
    - pipeline:
        name: "service-map-pipeline"

span-pipeline:
  source:
    pipeline:
      name: "entry-pipeline"
  processor:
    - otel_traces
  sink:
    - opensearch:
        ...
        index_type: trace-analytics-raw

service-map-pipeline:
  source:
    pipeline:
      name: "entry-pipeline"
  processor:
    - service_map
  sink:
    - opensearch:
        ...
        index_type: trace-analytics-service-map
```

Anda harus menjalankan OpenTelemetry Collector di lingkungan Anda untuk mengirim data ke titik akhir konsumsi. Untuk contoh pipeline lainnya, lihat cetak biru pipeline Trace Analytics. Untuk informasi selengkapnya, lihat the section called “Menggunakan cetak biru untuk membuat pipeline”.

Menurunkan metrik dari jejak dengan Amazon Ingestion OpenSearch

Anda dapat menggunakan Amazon OpenSearch Ingestion untuk mendapatkan metrik dari jejak. OpenTelemetry Contoh pipeline berikut menerima jejak masuk dan mengekstrak metrik yang disebut`durationInNanos`, dikumpulkan melalui jendela jatuh 30 detik. Kemudian mendapatkan histogram dari jejak yang masuk.

Pipeline berisi sub-pipeline berikut:

- `entry-pipeline`— Menerima data jejak dari OpenTelemetry kolektor dan meneruskannya ke `trace_to_metrics_pipeline` sub-pipeline.
- `trace-to-metrics-pipeline`— Menerima data jejak dari `entry-pipeline` sub-pipeline, menggabungkannya, dan memperoleh histogram `durationInNanos` dari jejak berdasarkan nilai bidang. `serviceName` Kemudian mengirimkan metrik turunan ke OpenSearch indeks yang disebut`metrics_for_traces`.

```
version: "2"
entry-pipeline:
  source:
    otel_trace_source:
      # Provide the path for ingestion. ${pipelineName} will be replaced with sub-
pipeline name.
      # In this case it would be "/entry-pipeline/v1/traces". This will be endpoint URI
 path in OpenTelemetry Exporter configuration.
      path: "/${pipelineName}/v1/traces"
  sink:
    - pipeline:
        name: "trace-to-metrics-pipeline"

trace-to-metrics-pipeline:
  source:
    pipeline:
      name: "entry-pipeline"
  processor:
    - aggregate:
        # Pick the required identification keys
        identification_keys: ["serviceName"]
        action:
          histogram:
            # Pick the appropriate values for each of the following fields
            key: "durationInNanos"
```

```
            record_minmax: true
            units: "seconds"
            buckets: [0, 10000000, 50000000, 100000000]
        # Specify an aggregation period
        group_duration: "30s"
sink:
  - opensearch:
        ...
        index: "metrics_for_traces"
```

Untuk contoh pipeline lainnya, lihat cetak biru jalur anomali Trace to metric anomali. Untuk informasi selengkapnya tentang cetak biru, lihat the section called “Menggunakan cetak biru untuk membuat pipeline”.

Deteksi anomali dengan Amazon Ingestion OpenSearch

Anda dapat menggunakan Amazon OpenSearch Ingestion untuk melatih model dan menghasilkan anomali hampir real-time pada peristiwa gabungan timeseries. Anda dapat menghasilkan anomali baik pada peristiwa yang dihasilkan dalam pipeline, atau pada peristiwa yang datang langsung ke pipeline, seperti OpenTelemetry metrik.

Anda dapat memasukkan peristiwa timeseries gabungan jendela jatuh ini ke prosesor detektor Anomali, yang melatih model dan menghasilkan anomali dengan skor nilai. Kemudian, tulis anomali ke indeks terpisah untuk membuat monitor dokumen dan memicu peringatan cepat.

Selain contoh-contoh ini, Anda juga dapat menggunakan pipeline anomali Log to metric dan Trace to metric anomaly pipeline blueprints. Untuk informasi selengkapnya tentang cetak biru, lihat the section called “Menggunakan cetak biru untuk membuat pipeline”.

Topik

- Metrik dari log
- Metrik dari jejak
- OpenTelemetry metrik

Metrik dari log

Pipeline berikut menerima log melalui sumber HTTP seperti FluentBit, mengekstrak nilai-nilai penting dari log dengan mencocokkan nilai dalam `log` kunci terhadap pola log Apache umum grok, dan

kemudian meneruskan log grokked ke kedua `log-to-metrics-pipeline` sub-pipeline, serta ke indeks bernama. OpenSearch `logs`

`log-to-metrics-pipeline`Sub-pipeline menerima log grokked dari `apache-log-pipeline-with-metrics` sub-pipeline, menggabungkannya, dan memperoleh metrik histogram berdasarkan nilai dalam dan kunci. `clientip` `request` Kemudian mengirimkan metrik histogram ke OpenSearch indeks bernama`histogram_metrics`, serta ke sub-pipeline. `log-to-metrics-anomaly-detector`

`log-to-metrics-anomaly-detector-pipeline`Sub-pipeline menerima metrik histogram agregat dari `log-to-metrics-pipeline` sub-pipeline dan mengirimkannya ke prosesor detektor Anomali untuk mendeteksi anomali menggunakan algoritma Random Cut Forest. Jika mendeteksi anomali, ia mengirimkannya ke indeks bernama OpenSearch . `log-metric-anomalies`

```
version: "2"
apache-log-pipeline-with-metrics:
  source:
    http:
      # Provide the path for ingestion. ${pipelineName} will be replaced with pipeline
 name configured for this pipeline.
      # In this case it would be "/apache-log-pipeline-with-metrics/logs". This will be
 the FluentBit output URI value.
      path: "/${pipelineName}/logs"
  processor:
    - grok:
        match:
          log: [ "%{COMMONAPACHELOG_DATATYPED}" ]
  sink:
    - opensearch:
        ...
        index: "logs"
    - pipeline:
        name: "log-to-metrics-pipeline"

log-to-metrics-pipeline:
  source:
    pipeline:
      name: "apache-log-pipeline-with-metrics"
  processor:
    - aggregate:
        # Specify the required identification keys
        identification_keys: ["clientip", "request"]
```

```
        action:
          histogram:
            # Specify the appropriate values for each the following fields
            key: "bytes"
            record_minmax: true
            units: "bytes"
            buckets: [0, 25000000, 50000000, 75000000, 100000000]
        # Pick the required aggregation period
        group_duration: "30s"
  sink:
    - opensearch:
        ...
        index: "histogram_metrics"
    - pipeline:
        name: "log-to-metrics-anomaly-detector-pipeline"

log-to-metrics-anomaly-detector-pipeline:
  source:
    pipeline:
      name: "log-to-metrics-pipeline"
  processor:
    - anomaly_detector:
        # Specify the key on which to run anomaly detection
        keys: [ "bytes" ]
        mode:
          random_cut_forest:
  sink:
    - opensearch:
        ...
        index: "log-metric-anomalies"
```

Metrik dari jejak

Anda dapat memperoleh metrik dari jejak dan menemukan anomali dalam metrik yang dihasilkan ini. Dalam contoh ini, `entry-pipeline` sub-pipeline menerima data jejak dari OpenTelemetry Kolektor dan meneruskannya ke sub-pipeline berikut:

- `span-pipeline`— Ekstrak bentang mentah dari jejak. Ini mengirimkan bentang mentah ke indeks apa pun yang OpenSearch diawali dengan. `otel-v1-apm-span`
- `service-map-pipeline`— Mengagregat dan menganalisisnya untuk membuat dokumen yang mewakili koneksi antar layanan. Ini mengirimkan dokumen-dokumen ini ke OpenSearch indeks

bernamaotel-v1-apm-service-map. Anda kemudian dapat melihat visualisasi peta layanan melalui plugin Trace Analytics untuk OpenSearch Dasbor.

- trace-to-metrics-pipeline—Mengagregat dan menurunkan metrik histogram dari jejak berdasarkan nilai. serviceName Kemudian mengirimkan metrik turunan ke OpenSearch indeks bernamametrics_for_traces, serta ke trace-to-metrics-anomaly-detector-pipeline sub-pipeline.

trace-to-metrics-anomaly-detector-pipelineSub-pipeline menerima metrik histogram agregat dari trace-to-metrics-pipeline dan mengirimkannya ke prosesor detektor Anomali untuk mendeteksi anomali menggunakan algoritma Random Cut Forest. Jika mendeteksi anomali, ia mengirimkannya ke indeks bernama OpenSearch. trace-metric-anomalies

```
version: "2"
entry-pipeline:
  source:
    otel_trace_source:
      # Provide the path for ingestion. ${pipelineName} will be replaced with pipeline
 name configured for this pipeline.
      # In this case it would be "/entry-pipeline/v1/traces". This will be endpoint URI
 path in OpenTelemetry Exporter
      # configuration.
      # path: "/${pipelineName}/v1/traces"
  processor:
    - trace_peer_forwarder:
  sink:
    - pipeline:
        name: "span-pipeline"
    - pipeline:
        name: "service-map-pipeline"
    - pipeline:
        name: "trace-to-metrics-pipeline"

span-pipeline:
  source:
    pipeline:
      name: "entry-pipeline"
  processor:
    - otel_trace_raw:
  sink:
    - opensearch:
        ...
```

```
        index_type: "trace-analytics-raw"

service-map-pipeline:
  source:
    pipeline:
      name: "entry-pipeline"
  processor:
    - service_map:
  sink:
    - opensearch:
        ...
        index_type: "trace-analytics-service-map"

trace-to-metrics-pipeline:
  source:
    pipeline:
      name: "entry-pipeline"
  processor:
    - aggregate:
        # Pick the required identification keys
        identification_keys: ["serviceName"]
        action:
          histogram:
            # Pick the appropriate values for each the following fields
            key: "durationInNanos"
            record_minmax: true
            units: "seconds"
            buckets: [0, 10000000, 50000000, 100000000]
        # Pick the required aggregation period
        group_duration: "30s"
  sink:
    - opensearch:
        ...
        index: "metrics_for_traces"
    - pipeline:
        name: "trace-to-metrics-anomaly-detector-pipeline"

trace-to-metrics-anomaly-detector-pipeline:
  source:
    pipeline:
      name: "trace-to-metrics-pipeline"
  processor:
    - anomaly_detector:
```

```
        # Below Key will find anomalies in the max value of histogram generated for
 durationInNanos.
        keys: [ "max" ]
        mode:
          random_cut_forest:
  sink:
    - opensearch:
        ...
        index: "trace-metric-anomalies"
```

OpenTelemetry metrik

Anda dapat membuat pipeline yang menerima OpenTelemetry metrik dan mendeteksi anomali dalam metrik ini. Dalam contoh ini, `entry-pipeline` menerima data metrik dari OpenTelemetry Kolektor. Jika metrik adalah tipe `GAUGE` dan nama metriknya`totalApiBytesSent`, prosesor mengirimkannya ke `ad-pipeline` sub-pipeline.

`ad-pipeline`Sub-pipeline menerima data metrik dari jalur masuk dan melakukan deteksi anomali pada nilai metrik menggunakan prosesor detektor Anomali.

```
entry-pipeline:
  source:
    otel_metrics_source:
  processor:
    - otel_metrics:
  route:
    - gauge_route: '/kind = "GAUGE" and /name = "totalApiBytesSent"'
  sink:
    - pipeline:
        name: "ad-pipeline"
        routes:
          - gauge_route
    - opensearch:
        ...
        index: "otel-metrics"

ad-pipeline:
  source:
    pipeline:
      name: "entry-pipeline"
    processor:
      - anomaly_detector:
        # Use "value" as the key on which anomaly detector needs to be run
```

```
        keys: [ "value" ]
        mode:
          random_cut_forest:
    sink:
      - opensearch:
        ...
        index: otel-metrics-anomalies
```

Selain contoh ini, Anda juga dapat menggunakan cetak biru jalur Trace to metric anomaly. Untuk informasi selengkapnya tentang cetak biru, lihat the section called “Menggunakan cetak biru untuk membuat pipeline”.

Pengambilan sampel dengan Amazon Ingestion OpenSearch

Amazon OpenSearch Ingestion menyediakan kemampuan pengambilan sampel berikut. Selain contoh-contoh ini, Anda juga dapat menggunakan cetak biru pengambilan sampel log Apache. Untuk informasi selengkapnya tentang cetak biru, lihat the section called “Menggunakan cetak biru untuk membuat pipeline”.

Topik

- Pengambilan sampel waktu
- Persentase pengambilan sampel
- Pengambilan sampel ekor

Pengambilan sampel waktu

Anda dapat menggunakan `rate_limiter` tindakan dalam prosesor Agregat untuk membatasi jumlah peristiwa yang dapat diproses per detik. Anda dapat memilih untuk menghentikan acara berlebih atau meneruskannya ke periode waktu berikutnya.

Dalam contoh ini, hanya 100 peristiwa per detik dengan kode status `200` dikirim ke wastafel dari alamat IP tertentu. Ini menjatuhkan semua peristiwa berlebih dari jendela waktu yang dikonfigurasi.

```
...
  processor:
   - aggregate:

        identification_keys: ["clientip"]
```

```
        action:

          rate_limiter:

            events_per_second: 100

            when_exceeds: drop
        when: "/status == 200"
...
```

Jika Anda malah mengatur `when_exceeds` opsi ke`block`, prosesor akan memproses peristiwa berlebih di jendela waktu berikutnya.

Persentase pengambilan sampel

Gunakan `percent_sampler` tindakan dalam prosesor Agregat untuk membatasi jumlah peristiwa yang dikirim ke wastafel. Semua kelebihan acara akan dijatuhkan.

Dalam contoh ini, hanya 20 persen peristiwa dengan kode status `200` dikirim ke wastafel dari alamat IP yang diberikan:

```
...
  processor:
  - aggregate:

        identification_keys: ["clientip"]
        duration :

        action:

          percent_sampler:

            percent: 20

        when: "/status == 200"
...
```

Pengambilan sampel ekor

Gunakan `tail_sampler` tindakan dalam prosesor Agregat untuk mengambil sampel peristiwa berdasarkan serangkaian kebijakan yang ditentukan. Tindakan ini menunggu agregasi selesai di seluruh periode agregasi yang berbeda berdasarkan periode tunggu yang dikonfigurasi. Ketika

agregasi selesai, dan jika cocok dengan kondisi kesalahan tertentu, itu dikirim ke wastafel. Jika tidak, hanya persentase peristiwa yang dikonfigurasi yang dikirim ke wastafel.

Contoh pipeline berikut mengirimkan semua OpenTelemetry jejak dengan status kondisi kesalahan 2 ke wastafel. Ini hanya mengirimkan 20% jejak yang tidak cocok dengan kondisi kesalahan ini ke wastafel.

```
...
  processor:
   - aggregate:

         identification_keys: ["traceId"]

         action:

           tail_sampler:

             percent: 20

             wait_period: "10s"

             condition: "/status == 2"


...
```

Jika Anda menyetel kondisi kesalahan ke `false` atau tidak menyertakannya, hanya persentase peristiwa yang dikonfigurasi yang diizinkan untuk dilewati, ditentukan oleh hasil probabilistik.

Karena sulit untuk menentukan dengan tepat kapan pengambilan sampel ekor harus terjadi, Anda dapat menggunakan `wait_period` opsi untuk mengukur waktu idle setelah acara terakhir diterima.

Unduhan selektif dengan Amazon Ingestion OpenSearch

Jika pipeline Anda menggunakan sumber S3, Anda dapat menggunakan ekspresi SQL untuk melakukan pemfilteran dan perhitungan pada konten objek S3 sebelum memasukkannya ke dalam pipeline.

`s3_select`Opsi ini mendukung objek dalam format Parket. Ia juga bekerja dengan objek yang dikompresi dengan GZIP atau BZIP2 (hanya untuk objek CSV dan JSON), dan mendukung kompresi kolumnar untuk Parket menggunakan GZIP dan Snappy.

Contoh pipeline berikut mengunduh data dalam objek S3 yang masuk, dikodekan dalam format Parket:

```
pipeline:
  source:
    s3:
      s3_select:
        expression: "select * from s3object s"
        input_serialization: parquet
      notification_type: "sqs"
...
```

Contoh berikut hanya mengunduh 10.000 catatan pertama dalam objek:

```
pipeline:
  source:
    s3:
      s3_select:
        expression: "select * from s3object s LIMIT 10000"
        input_serialization: parquet
      notification_type: "sqs"
...
```

Contoh berikut memeriksa nilai minimum dan maksimum `data_value` sebelum memasukkan peristiwa ke dalam pipeline:

```
pipeline:
  source:
    s3:
      s3_select:
        expression: "select s.* from s3object s where s.data_value > 200 and
 s.data_value < 500 "
        input_serialization: parquet
      notification_type: "sqs"
...
```

Selain contoh-contoh ini, Anda juga dapat menggunakan cetak biru pipeline pilih S3. Untuk informasi selengkapnya tentang cetak biru, lihat the section called “Menggunakan cetak biru untuk membuat pipeline”.

Untuk informasi selengkapnya, silakan lihat sumber daya berikut:

- Memfilter dan mengambil data menggunakan Amazon S3 Select
- Referensi SQL untuk Amazon S3 Pilih

Keamanan di Amazon OpenSearch Ingestion

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 menggambarkan ini sebagai keamanan dari cloud dan keamanan di dalam cloud:

- Keamanan cloud – AWS bertanggung jawab untuk melindungi infrastruktur yang menjalankan layanan AWS di dalam AWS Cloud. AWS juga memberi layanan yang dapat Anda gunakan dengan aman. Auditor pihak ketiga menguji dan memverifikasi efektivitas keamanan kami sebagai bagian dari program kepatuhan AWS.
- Keamanan di cloud – Tanggung jawab Anda ditentukan menurut layanan AWS yang Anda gunakan. Anda juga bertanggung jawab atas faktor lain termasuk sensitivitas data Anda, persyaratan perusahaan Anda, serta hukum dan peraturan yang berlaku.

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

Topik

- Mengamankan saluran pipa Amazon OpenSearch Ingestion dalam VPC
- Identity and Access Management untuk Amazon OpenSearch Ingestion
- Menggunakan log panggilan API Amazon OpenSearch Ingestion AWS CloudTrail

Mengamankan saluran pipa Amazon OpenSearch Ingestion dalam VPC

Anda dapat meluncurkan saluran Amazon OpenSearch Ingestion ke cloud pribadi virtual (VPC). VPC adalah jaringan virtual yang didedikasikan untuk Akun AWSAnda. Ini secara logis terisolasi dari jaringan virtual lain di AWS Cloud. Menempatkan pipa di dalam VPC memungkinkan komunikasi

yang aman antara OpenSearch Ingestion dan layanan lain dalam VPC tanpa memerlukan gateway internet, perangkat NAT, atau koneksi VPN. Semua lalu lintas tetap aman di dalam AWS Cloud.

Menggunakan VPC memungkinkan Anda untuk menegakkan aliran data melalui pipa OpenSearch Ingestion Anda dalam batas-batas VPC, bukan melalui internet publik. Saluran pipa yang tidak berada dalam VPC mengirim dan menerima data melalui titik akhir yang menghadap publik dan internet.

Untuk instruksi untuk menyediakan pipeline dalam VPC, lihat. the section called “Membuat jaringan pipa”

Topik

Pertimbangan

Pertimbangkan hal berikut saat Anda mengonfigurasi akses VPC untuk pipeline.

- Pipeline publik dapat menulis ke domain VPC. Demikian pula, pipeline VPC dapat menulis ke domain publik.
- Pipeline tidak perlu berada di VPC yang sama dengan sink domainnya. Anda juga tidak perlu membuat koneksi antara dua VPC. OpenSearch Ingestion menangani menghubungkan mereka untuk Anda.
- Anda hanya dapat menentukan satu VPC untuk pipeline Anda.
- Berbeda dengan jaringan pipa publik, pipeline VPC harus Wilayah AWS sama dengan domain yang ditulisnya.
- Anda dapat memilih untuk menyebarkan pipeline ke dalam satu, dua, atau tiga subnet VPC Anda. Subnet didistribusikan di seluruh Availability Zone yang sama dengan Ingestion OpenSearch Compute Units (OCU) Anda digunakan.
- Jika Anda hanya menerapkan pipeline di satu subnet dan Availability Zone turun, Anda tidak akan dapat menyerap data. Untuk memastikan ketersediaan tinggi, kami sarankan Anda mengonfigurasi saluran pipa dengan dua atau tiga subnet.

- Menentukan grup keamanan adalah opsional. Jika Anda tidak menyediakan grup keamanan, kami menggunakan grup keamanan default yang ditentukan dalam VPC.

Batasan

Saluran pipa dalam VPC memiliki keterbatasan sebagai berikut.

- Anda tidak dapat mengubah konfigurasi jaringan pipeline setelah Anda membuatnya. Jika Anda meluncurkan pipeline dalam VPC, Anda nantinya tidak dapat mengubahnya menjadi titik akhir publik, dan sebaliknya.
- Anda dapat meluncurkan pipeline dalam VPC atau menggunakan titik akhir publik, tetapi Anda tidak dapat melakukan keduanya. Anda harus memilih satu atau yang lain ketika Anda membuat pipa.
- Setelah menyediakan pipeline dalam VPC, Anda tidak dapat memindahkannya ke VPC lain, dan Anda tidak dapat mengubah subnet atau pengaturan grup keamanannya.
- Jika pipeline Anda menulis ke sink domain VPC, Anda tidak dapat kembali lagi nanti dan mengubah wastafel ke domain lain (VPC atau publik) setelah pipeline dibuat. Anda harus menghapus dan membuat ulang pipa dengan wastafel baru. Anda masih dapat mengalihkan wastafel dari domain publik ke domain VPC.
- Anda tidak dapat memberikan akses konsumsi lintas akun ke pipeline VPC.

Prasyarat

Sebelum Anda dapat menyediakan pipeline dalam VPC, Anda harus melakukan hal berikut:

- Buat VPC

  Untuk membuat VPC, Anda dapat menggunakan konsol VPC Amazon, AWS CLI, atau salah satu SDK. AWS Untuk informasi lebih lanjut, lihat Bekerja dengan VPC di Panduan Pengguna Amazon VPC. Jika Anda sudah memiliki VPC, Anda dapat melewati langkah ini.

- Cadangan alamat IP

  OpenSearch Ingestion menempatkan elastic network interface di setiap subnet yang Anda tentukan selama pembuatan pipeline. Setiap antarmuka jaringan dikaitkan dengan alamat IP. Anda harus memesan satu alamat IP per subnet untuk antarmuka jaringan.

Mengkonfigurasi akses VPC untuk pipeline

Anda dapat mengaktifkan akses VPC untuk pipeline di dalam konsol OpenSearch Layanan atau menggunakan. AWS CLI

Konsol

Anda mengonfigurasi akses VPC selama pembuatan pipeline. Di bawah Jaringan, pilih akses VPC dan konfigurasikan pengaturan berikut:

Pengaturan	Deskripsi
VPC	Pilih ID virtual private cloud (VPC) yang ingin Anda gunakan. VPC dan pipeline harus sama. Wilayah AWS
Subnet	Pilih satu atau lebih subnet. OpenSearch Layanan akan menempatkan titik akhir VPC dan antarmuka jaringan elastis di subnet.
Grup keamanan	Pilih satu atau beberapa grup keamanan VPC yang memungkinkan aplikasi Anda mencapai pipeline OpenSearch Ingestion pada port (80 atau 443) dan protokol (HTTP atau HTTPs) yang diekspos oleh pipeline.

CLI

Untuk mengkonfigurasi akses VPC menggunakan AWS CLI, tentukan parameter: `--vpc-options`

```
aws osis create-pipeline \
  --pipeline-name vpc-pipeline \
  --min-units 4 \
  --max-units 10 \
  --vpc-options
 SecurityGroupIds={sg-12345678,sg-9012345},SubnetIds=subnet-1212234567834asdf \
  --pipeline-configuration-body "file://pipeline-config.yaml"
```

Peran yang terhubung dengan layanan untuk akses VPC

Peran yang terhubung dengan layanan adalah tipe IAM role unik yang mendelegasikan izin untuk layanan sehingga dapat membuat dan mengelola sumber daya atas nama Anda. OpenSearch Ingestion memerlukan peran terkait layanan yang dipanggil untuk AWSServiceRoleForAmazonOpenSearchIngestionmengakses VPC Anda, membuat titik akhir

pipeline, dan menempatkan antarmuka jaringan di subnet VPC Anda. Untuk informasi selengkapnya tentang izin peran ini dan cara menghapusnya, lihatthe section called “Peran pembuatan pipa”.

OpenSearch Ingestion secara otomatis membuat peran saat Anda membuat pipeline konsumsi. Agar pembuatan otomatis ini berhasil, pengguna yang membuat pipeline pertama di akun harus memiliki izin untuk `iam:CreateServiceLinkedRole` tindakan tersebut. Untuk mempelajari selengkapnya, lihat Izin peran terkait layanan di Panduan Pengguna IAM. Anda dapat melihat peran di konsol AWS Identity and Access Management (IAM) setelah dibuat.

Identity and Access Management untuk Amazon OpenSearch Ingestion

AWS Identity and Access Management(IAM) adalah Layanan AWS yang membantu administrator mengontrol akses ke AWS sumber daya secara aman. Administrator IAM mengontrol siapa yang dapat terautentikasi (masuk) dan berwenang (memiliki izin) untuk menggunakan OpenSearch sumber daya Tertelan. IAM adalah layanan Layanan AWS yang dapat Anda gunakan tanpa biaya tambahan.

Topik

- Kebijakan berbasis identitas untuk Tertelan OpenSearch
- Tindakan kebijakan untuk OpenSearch Tertelan
- Sumber daya kebijakan untuk OpenSearch Penyerapan
- Kunci ketentuan kebijakan untuk Amazon OpenSearch Ingestion
- ABAC dengan OpenSearch Tertelan
- Menggunakan kredensi sementara dengan Tertelan OpenSearch
- Peran tertaut layanan untuk Tertelan OpenSearch
- Contoh kebijakan berbasis identitas untuk Tertelan OpenSearch

Kebijakan berbasis identitas untuk Tertelan OpenSearch

Hanya mendukung kebijakan berbasis identitas	Ya

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

Dengan kebijakan berbasis identitas IAM, Anda dapat menentukan tindakan dan sumber daya yang diizinkan atau ditolak, serta kondisi di mana tindakan tersebut diperbolehkan atau ditolak. Anda tidak dapat menentukan pelaku utama dalam kebijakan berbasis identitas karena itu berlaku untuk pengguna atau peran yang dilampiri kebijakan. Untuk mempelajari semua elemen yang dapat Anda gunakan dalam kebijakan JSON, lihat Referensi elemen kebijakan IAM JSON dalam Panduan Pengguna IAM.

Contoh kebijakan berbasis identitas untuk Tertelan OpenSearch

Untuk melihat contoh kebijakan berbasis identitas OpenSearch tertelan, lihat. the section called "Contoh kebijakan berbasis identitas"

Tindakan kebijakan untuk OpenSearch Tertelan

Mendukung tindakan kebijakan	Ya

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

Sertakan tindakan dalam kebijakan untuk memberikan izin guna melakukan operasi yang terkait.

Tindakan kebijakan di OpenSearch Penyerapan menggunakan prefiks berikut sebelum tindakan:

```
osis
```

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

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

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

```
"Action": "osis:List*"
```

Untuk melihat contoh kebijakan berbasis identitas OpenSearch tertelan, lihat. Contoh kebijakan berbasis identitas untuk Tanpa Server OpenSearch

Sumber daya kebijakan untuk OpenSearch Penyerapan

Mendukung sumber daya kebijakan	Ya

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

Elemen kebijakan JSON `Resource` menentukan objek atau objek-objek yang menjadi target penerapan tindakan. Pernyataan harus mencakup elemen `Resource` atau `NotResource`. Sebagai praktik terbaik, tentukan sumber daya menggunakan Amazon Resource Name (ARN). Anda dapat melakukan ini untuk tindakan yang mendukung tipe sumber daya tertentu, yang dikenal sebagai izin tingkat sumber daya.

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

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

Kunci ketentuan kebijakan untuk Amazon OpenSearch Ingestion

Mendukung kunci kondisi kebijakan khusus layanan	Tidak

Administrator dapat menggunakan kebijakan JSON AWS untuk menentukan siapa yang memiliki akses ke hal apa. Yaitu, prinsipal mana yang dapat melakukan tindakan pada sumber daya apa, dan menurut persyaratan apa.

Elemen `Condition` (atau `Condition` blok) memungkinkan Anda menentukan syarat di mana suatu pernyataan berlaku. Elemen `Condition` bersifat opsional. Anda dapat membuat ekspresi bersyarat

yang menggunakan operator syarat, seperti sama dengan atau kurang dari, untuk mencocokkan syarat dalam kebijakan dengan nilai dalam permintaan.

Jika Anda menentukan beberapa elemen `Condition` dalam pernyataan, atau beberapa kunci dalam satu elemen `Condition`, AWS akan mengevaluasinya denga menggunakan operasi logika `AND`. Jika Anda menetapkan beberapa nilai untuk kunci syarat tunggal, AWS akan mengevaluasi syarat tersebut dengan menggunakan operasi logika `OR`. Semua persyaratan harus dipenuhi sebelum izin pernyataan diberikan.

Anda juga dapat menggunakan variabel placeholder saat menentukan syarat. Sebagai contoh, Anda dapat memberikan izin pengguna IAM untuk mengakses sumber daya hanya jika ditandai dengan nama pengguna IAM mereka. Untuk informasi lebih lanjut, lihat Elemen kebijakan IAM: variabel dan tag dalam Panduan Pengguna IAM.

AWS mendukung kunci syarat global dan kunci syarat khusus layanan. Untuk melihat semua kunci syarat global AWS, lihat Kunci konteks syarat global AWS dalam Panduan Pengguna IAM.

Untuk melihat daftar kunci syarat OpenSearch konsumsi, lihat Kunci syarat untuk Amazon OpenSearch Ingestion dalam Referensi Otorisasi Layanan. Untuk mempelajari dengan tindakan dan sumber daya mana Anda dapat menggunakan kunci syarat, lihat Tindakan yang ditentukan oleh Amazon OpenSearch Ingestion.

ABAC dengan OpenSearch Tertelan

Mendukung ABAC (tanda dalam kebijakan)	Ya

Kontrol akses berbasis atribut (ABAC) adalah strategi otorisasi yang mendefinisikan izin berdasarkan atribut. Di AWS, atribut ini disebut tanda. Anda dapat melampirkan tanda ke entitas IAM (pengguna atau peran) dan ke banyak sumber daya AWS. Penandaan entitas dan sumber daya adalah langkah pertama dari ABAC. Kemudian Anda merancang kebijakan ABAC untuk mengizinkan operasi ketika tanda pelaku utama cocok dengan tanda di sumber daya yang ingin diakses.

ABAC sangat membantu di lingkungan yang berkembang dengan cepat dan membantu dalam situasi ketika manajemen kebijakan menjadi rumit.

Untuk mengontrol akses berdasarkan tandanya, Anda memberikan informasi tanda di elemen syarat kebijakan dengan menggunakan kunci syarat `aws:ResourceTag/`*`key-name`*, `aws:RequestTag/`*`key-name`*, atau `aws:TagKeys`.

Jika layanan mendukung ketiga kunci kondisi untuk setiap jenis sumber daya, maka nilainya adalah Ya untuk layanan. Jika layanan mendukung ketiga kunci kondisi hanya untuk beberapa jenis sumber daya, maka nilainya adalah Partial.

Untuk informasi lebih lanjut tentang ABAC, lihat Apa itu ABAC? dalam Panduan Pengguna IAM. Untuk melihat tutorial dengan langkah-langkah untuk menyiapkan ABAC, lihat Menggunakan kontrol akses berbasis atribut (ABAC) dalam Panduan Pengguna IAM.

Untuk informasi lebih lanjut tentang penandaan sumber daya OpenSearch Inestion, lihat. the section called “Menandai”

Menggunakan kredensi sementara dengan Tertelan OpenSearch

Mendukung penggunaan kredensial sementara	Ya

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

Anda menggunakan kredensial sementara jika Anda masuk ke AWS Management Console menggunakan metode apa pun kecuali nama pengguna dan kata sandi. Misalnya, saat Anda mengakses AWS dengan menggunakan tautan masuk tunggal (SSO) perusahaan Anda, proses tersebut akan membuat kredensial sementara secara otomatis. Anda juga secara otomatis membuat kredensial sementara ketika Anda masuk ke konsol sebagai pengguna dan kemudian beralih peran. Untuk informasi selengkapnya tentang beralih peran, lihat Beralih ke peran (konsol) dalam Panduan Pengguna IAM.

Anda dapat secara manual membuat kredensial sementara menggunakan AWS CLI atau API AWS. Anda kemudian dapat menggunakan kredensial sementara tersebut untuk mengakses AWS. AWS menyarankan agar Anda membuat kredensial sementara secara dinamis alih-alih menggunakan access key jangka panjang. Untuk informasi lebih lanjut, lihat Kredensial keamanan sementara di IAM.

Peran tertaut layanan untuk Tertelan OpenSearch

Mendukung peran yang terhubung dengan layanan	Ya

Peran tertaut layanan adalah jenis peran layanan yang tertaut dengan layanan yang tertaut dengan. Layanan AWS Layanan dapat menggunakan peran untuk melakukan tindakan atas nama Anda. Peran tertaut layanan muncul di peran tertaut layanan muncul di Anda Akun AWS dan dimiliki oleh layanan tersebut. Administrator IAM dapat melihat, tetapi tidak dapat mengedit izin untuk peran terkait layanan.

OpenSearchTertelan menggunakan peran terkait layanan yang disebut. `AWSServiceRoleForAmazonOpenSearchIngestion` Untuk detail tentang membuat dan mengelola peran tertaut layanan OpenSearch tertaut-layanan, lihat. the section called “Peran pembuatan pipa”

Contoh kebijakan berbasis identitas untuk Tertelan OpenSearch

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

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

Untuk detail tentang tindakan dan jenis sumber daya yang ditentukan oleh Amazon OpenSearch Ingestion, termasuk format ARN untuk setiap jenis sumber daya, lihat Tindakan, sumber daya, dan kunci kondisi untuk Amazon OpenSearch Ingestion dalam Referensi Otorisasi Layanan.

Topik

- Praktik terbaik kebijakan
- Menggunakan OpenSearch Tertelan di konsol
- Mengelola saluran OpenSearch pipa konsumsi
- Menelan data ke dalam pipeline Ingestion OpenSearch

Praktik terbaik kebijakan

Kebijakan berbasis identitas adalah pilihan yang sangat tepat. Kebijakan ini menentukan apakah seseorang dapat membuat, mengakses, atau menghapus sumber daya OpenSearch Tertelan di akun

Anda. Tindakan ini membuat Akun AWS Anda terkena biaya. Ketika Anda membuat atau mengedit kebijakan berbasis identitas, ikuti panduan dan rekomendasi ini:

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

- Memulai kebijakan AWS terkelola dan beralih ke izin paling sedikit hak istimewa — Untuk mulai memberikan izin kepada pengguna dan beban kerja Anda, gunakan kebijakan AWS terkelola yang memberikan izin untuk banyak kasus penggunaan umum. Mereka tersedia di AndaAkun AWS. Kami menyarankan Anda mengurangi izin lebih lanjut dengan menentukan kebijakan yang dikelola AWS pelanggan yang spesifik untuk kasus penggunaan Anda. Untuk informasi selengkapnya, lihat kebijakan AWSAWS terkelola atau kebijakan terkelola untuk fungsi pekerjaan di Panduan Pengguna IAM.
- Menerapkan izin hak akses terkecil — Saat Anda menetapkan izin dengan kebijakan IAM, berikan izin yang diperlukan untuk melaksanakan tugas. Anda melakukan ini dengan menentukan tindakan yang dapat diambil pada sumber daya tertentu dalam kondisi tertentu, juga dikenal sebagai izin paling tidak memiliki hak istimewa. Untuk informasi selengkapnya tentang penggunaan IAM untuk menerapkan izin, lihat Kebijakan dan izin di IAM dalam Panduan Pengguna IAM.
- Gunakan ketentuan dalam kebijakan IAM untuk membatasi akses lebih lanjut — Anda dapat menambahkan kondisi pada kebijakan Anda untuk membatasi akses ke tindakan dan sumber daya. Misalnya, Anda dapat menulis ketentuan kebijakan untuk menentukan bahwa semua permintaan harus dikirim menggunakan SSL. Anda juga dapat menggunakan kondisi untuk memberikan akses ke tindakan layanan jika digunakan melalui spesifikLayanan AWS, sepertiAWS CloudFormation. Untuk informasi lebih lanjut, lihat Elemen Kebijakan IAM JSON: Syarat dalam Panduan Pengguna IAM.
- Gunakan IAM Access Analyzer untuk memvalidasi kebijakan IAM Anda untuk memastikan izin yang aman dan fungsional - IAM Access Analyzer memvalidasi kebijakan baru dan yang sudah ada sehingga kebijakan mematuhi bahasa kebijakan IAM (JSON) dan praktik terbaik IAM. IAM Access Analyzer menyediakan lebih dari 100 pemeriksaan kebijakan dan rekomendasi yang dapat ditindaklanjuti untuk membantu Anda membuat kebijakan yang aman dan fungsional. Untuk informasi selengkapnya, lihat validasi kebijakan IAM Access Analyzer di Panduan Pengguna IAM.
- Memerlukan otentikasi multi-faktor (MFA) — Jika Anda memiliki skenario yang mengharuskan pengguna IAM atau pengguna root di AndaAkun AWS, aktifkan MFA untuk keamanan tambahan. Untuk mewajibkan MFA saat operasi API dipanggil, tambahkan kondisi MFA ke kebijakan Anda.

Untuk informasi selengkapnya, lihat Mengonfigurasi akses API yang dilindungi MFA di Panduan Pengguna IAM.

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

Menggunakan OpenSearch Tertelan di konsol

Untuk mengakses OpenSearch Penyerapan dalam konsol OpenSearch Layanan, Anda harus memiliki rangkaian izin minimum. Izin ini harus memperbolehkan Anda untuk membuat daftar dan melihat detail tentang sumber daya OpenSearch Tertelan di akun Anda. AWS Jika Anda membuat kebijakan berbasis identitas yang lebih ketat dari izin minimum yang diperlukan, konsol tidak akan berfungsi sebagaimana dimaksudkan untuk entitas (seperti peran IAM) dengan kebijakan tersebut.

Anda tidak perlu mengizinkan konsol minimum untuk pengguna yang melakukan panggilan hanya ke AWS CLI atau API AWS. Alih-alih, izinkan akses hanya ke tindakan yang sesuai dengan operasi API yang Anda coba lakukan.

Kebijakan berikut memungkinkan pengguna untuk mengakses OpenSearch Penyerapan dalam konsol OpenSearch Layanan:

```
{
    "Version": "2012-10-17",
    "Statement": [
        {
            "Resource": "*",
            "Effect": "Allow",
            "Action": [
                "osis:ListPipelines",
                "osis:GetPipeline",
                "osis:ListPipelineBlueprints",
                "osis:GetPipelineBlueprint",
                "osis:GetPipelineChangeProgress"
            ]
        }
    ]
}
```

Sebagai alternatif, Anda dapat menggunakan kebijakan the section called “AmazonOpenSearchIngestionReadOnlyAccess” AWS terkelola, yang memberikan akses hanya-baca ke semua OpenSearch sumber daya Penyerapan untuk sebuah. Akun AWS

Mengelola saluran OpenSearch pipa konsumsi

Kebijakan ini adalah contoh kebijakan “admin pipeline” yang memungkinkan pengguna mengelola dan mengelola pipeline Amazon OpenSearch Ingestion. Pengguna dapat membuat, melihat, dan menghapus jaringan pipa.

```
{
    "Version": "2012-10-17",
    "Statement": [
        {
            "Resource": "arn:aws:osis:region:123456789012:pipeline/*",
            "Action": [
                "osis:CreatePipeline",
                "osis:DeletePipeline",
                "osis:UpdatePipeline",
                "osis:ValidatePipeline",
                "osis:StartPipeline",
                "osis:StopPipeline"
            ],
            "Effect": "Allow"
        },
        {
            "Resource": "*",
            "Action": [
                "osis:ListPipelines",
                "osis:GetPipeline",
                "osis:ListPipelineBlueprints",
                "osis:GetPipelineBlueprint",
                "osis:GetPipelineChangeProgress"
            ],
            "Effect": "Allow"
        }
    ]
}
```

Menelan data ke dalam pipeline Ingestion OpenSearch

Kebijakan contoh ini memungkinkan pengguna atau entitas lain untuk menyerap data ke dalam pipeline Amazon OpenSearch Ingestion di akun mereka. Pengguna tidak dapat memodifikasi saluran pipa.

```
{
```

```
    "Version": "2012-10-17",
    "Statement": [
        {
            "Resource": "arn:aws:osis:region:123456789012:pipeline/*",
            "Action": [
                "osis:Ingest"
            ],
            "Effect": "Allow"
        }
    ]
}
```

Menggunakan log panggilan API Amazon OpenSearch Ingestion AWS CloudTrail

Amazon OpenSearch Ingestion terintegrasi denganAWS CloudTrail, layanan yang menyediakan catatan tindakan yang dilakukan oleh pengguna, peran, atau AWS layanan di OpenSearch Inestion.

CloudTrailmerekam semua panggilan API untuk OpenSearch menelan sebagai kejadian. Panggilan yang direkam mencakup panggilan dari bagian OpenSearch Ingestion pada konsol OpenSearch Layanan dan panggilan kode ke operasi API OpenSearch Ingestion.

Jika membuat jejak, Anda dapat mengaktifkan pengiriman berkelanjutan dari CloudTrail bucket Amazon S3, termasuk kejadian untuk OpenSearch Inestion. Jika Anda tidak membuat konfigurasi jejak, Anda masih dapat melihat kejadian terbaru dalam konsol CloudTrail di Riwayat peristiwa.

Menggunakan informasi yang dikumpulkan olehCloudTrail, Anda dapat menentukan permintaan yang dibuat ke OpenSearch Inestion, alamat IP asal permintaan tersebut dibuat, siapa yang membuat permintaan, kapan permintaan dibuat, dan detail lainnya.

Untuk mempelajari lebih lanjutCloudTrail, lihat Panduan AWS CloudTrail Pengguna.

OpenSearchInformasi menelan di CloudTrail

CloudTraildiaktifkan pada Anda Akun AWS saat Anda membuat akun. Ketika aktivitas terjadi di OpenSearch Inestion, aktivitas tersebut dicatat dalam CloudTrail peristiwa bersama peristiwa AWS layanan lainnya di Riwayat peristiwa. Anda dapat melihat, mencari, dan mengunduh peristiwa terbaru di Akun AWS Anda. Untuk informasi lebih lanjut, lihat CloudTrail peristiwa dengan riwayat tindakan.

Untuk catatan berkelanjutan tentang peristiwa di AndaAkun AWS, termasuk peristiwa untuk OpenSearch Tertelan, buat jejak. Jejak memungkinkan CloudTrail untuk mengirim berkas log ke

bucket Amazon S3. Secara default, saat Anda membuat jejak di konsol, jejak tersebut berlaku untuk semua Wilayah AWS.

Jejak mencatat peristiwa dari semua Wilayah di partisi AWS dan mengirimkan berkas log ke bucket Amazon S3 yang Anda tentukan. Selain itu, Anda dapat membuat konfigurasi layanan AWS lainnya untuk menganalisis lebih lanjut dan bertindak berdasarkan data peristiwa yang dikumpulkan di log CloudTrail. Untuk informasi selengkapnya, lihat yang berikut:

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

Semua tindakan OpenSearch Penyerapan dicatat oleh CloudTrail dan didokumentasikan dalam referensi API OpenSearchIngestion. Misalnya, panggilan untuk tindakan `CreateCollection`, `ListCollections`, dan `DeleteCollection` menghasilkan entri dalam file log CloudTrail.

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

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

Untuk informasi selengkapnya, lihat Elemen userIdentity CloudTrail.

OpenSearchMemahami entri file log

Jejak adalah konfigurasi yang memungkinkan pengiriman peristiwa sebagai berkas log ke bucket Amazon S3 yang telah Anda tentukan. File log CloudTrail berisi satu atau lebih entri log.

Suatu peristiwa mewakili permintaan tunggal dari semua sumber. Peristiwa ini mencakup informasi tentang tindakan yang diminta, tanggal dan waktu tindakan, parameter permintaan, dan sebagainya. Berkas log CloudTrail bukan jejak tumpukan terurut dari panggilan API publik, sehingga berkas tersebut tidak muncul dalam urutan tertentu.

Contoh berikut menunjukkan CloudTrail catatan log yang menunjukkan `DeletePipeline` tindakan.


```
{
    "eventVersion": "1.08",
    "userIdentity": {
        "type": "AssumedRole",
        "principalId": "AIDACKCEVSQ6C2EXAMPLE",
        "arn":"arn:aws:iam::123456789012:user/test-user",
        "accountId": "123456789012",
        "accessKeyId": "access-key",
        "sessionContext": {
            "sessionIssuer": {
                "type": "Role",
                "principalId": "AIDACKCEVSQ6C2EXAMPLE",
                "arn": "arn:aws:iam::123456789012:role/Admin",
                "accountId": "123456789012",
                "userName": "Admin"
            },
            "webIdFederationData": {},
            "attributes": {
                "creationDate": "2023-04-21T16:48:33Z",
                "mfaAuthenticated": "false"
            }
        }
    },
    "eventTime": "2023-04-21T16:49:22Z",
    "eventSource": "osis.amazonaws.com",
    "eventName": "UpdatePipeline",
    "awsRegion": "us-west-2",
    "sourceIPAddress": "123.456.789.012",
    "userAgent": "Mozilla/5.0 (Macintosh; Intel Mac OS X 10_15_7) AppleWebKit/537.36
 (KHTML, like Gecko) Chrome/112.0.0.0 Safari/537.36",
    "requestParameters": {
        "pipelineName": "my-pipeline",
        "pipelineConfigurationBody": "version: \"2\"\nlog-pipeline:\n  source:\n
 http:\n        path: \"/test/logs\"\n  processor:\n    - grok:\n        match:\n
        log: [ '%{COMMONAPACHELOG}' ]\n    - date:\n        from_time_received: true
\n        destination: \"@timestamp\"\n  sink:\n    - opensearch:\n        hosts:
 [ \"https://search-b5zd22mwxhggheqpj5ftslgyle.us-west-2.es.amazonaws.com\" ]\n
  index: \"apache_logs2\"\n        aws_sts_role_arn: \"arn:aws:iam::709387180454:role/
canary-bootstrap-OsisRole-J1BARLD26QKN\"\n        aws_region: \"us-west-2\"\n
 aws_sigv4: true\n"
    },
    "responseElements": {
```

```
        "pipeline": {
            "pipelineName": "my-pipeline",sourceIPAddress
            "pipelineArn": "arn:aws:osis:us-west-2:123456789012:pipeline/my-pipeline",
            "minUnits": 1,
            "maxUnits": 1,
            "status": "UPDATING",
            "statusReason": {
                "description": "An update was triggered for the pipeline. It is still
 available to ingest data."
            },
            "pipelineConfigurationBody": "version: \"2\"\nlog-pipeline:\n  source:\n
    http:\n        path: \"/test/logs\"\n  processor:\n    - grok:\n        match:
\n          log: [ '%{COMMONAPACHELOG}' ]\n    - date:\n        from_time_received:
 true\n        destination: \"@timestamp\"\n  sink:\n    - opensearch:\n        hosts:
 [ \"https://search-b5zd22mwxhggheqpj5ftslgyle.us-west-2.es.amazonaws.com\" ]\n
  index: \"apache_logs2\"\n        aws_sts_role_arn: \"arn:aws:iam::709387180454:role/
canary-bootstrap-OsisRole-J1BARLD26QKN\"\n        aws_region: \"us-west-2\"\n
 aws_sigv4: true\n",
            "createdAt": "Mar 29, 2023 1:03:44 PM",
            "lastUpdatedAt": "Apr 21, 2023 9:49:21 AM",
            "ingestEndpointUrls": [
                "my-pipeline-tu33ldsgdltgv7x7tjqiudvf7m.us-west-2.osis.amazonaws.com"
            ]
        }
    },
    "requestID": "12345678-1234-1234-1234-987654321098",
    "eventID": "12345678-1234-1234-1234-987654321098",
    "readOnly": false,
    "eventType": "AwsApiCall",
    "managementEvent": true,
    "recipientAccountId": "709387180454",
    "eventCategory": "Management",
    "tlsDetails": {
        "tlsVersion": "TLSv1.2",
        "cipherSuite": "ECDHE-RSA-AES128-GCM-SHA256",
        "clientProvidedHostHeader": "osis.us-west-2.amazonaws.com"
    },
    "sessionCredentialFromConsole": "true"
}
```

Menandai jaringan pipa Amazon OpenSearch Ingestion

Tanda memungkinkan Anda menugaskan informasi arbitrer ke alur Amazon OpenSearch sehingga Anda dapat mengkategorikan dan mem-filter informasi tersebut. Tanda adalah label metadata yang Anda tetapkan atau AWS yang ditetapkan ke sumber daya AWS. Setiap tanda terdiri dari kunci dan nilai. Untuk tanda yang Anda tetapkan, Anda menentukan kunci dan nilai. Misalnya, Anda dapat menentukan kunci sebagai `stage` dan nilai untuk satu sumber daya sebagai `test`.

Tanda membantu Anda melakukan hal berikut:

- Identifikasi dan organisir sumber daya AWS Anda. Banyak layanan AWS yang mendukung penandaan, sehingga Anda dapat menetapkan tanda yang sama ke sumber daya dari layanan yang berbeda untuk menunjukkan bahwa sumber daya tersebut terkait. Contohnya, Anda dapat menugaskan tanda yang sama ke alur OpenSearch Ingestion yang Anda tetapkan ke domain Amazon OpenSearch Service.
- Telusuri biaya AWS Anda. Anda mengaktifkan tag ini pada AWS Billing and Cost Management dasbor.AWS menggunakan tag untuk mengategorikan biaya Anda lalu mengirimkan laporan alokasi biaya bulanan kepada Anda. Untuk informasi selengkapnya, lihat Gunakan Tag Alokasi Biaya dalam AWS BillingPanduan Pengguna.
- Batasi akses ke jaringan pipa menggunakan kontrol akses berbasis atribut. Untuk informasi lebih lanjut, lihat Mengendalikan akses berdasarkan kunci tanda di Panduan Pengguna IAM.

Dalam OpenSearch menelan, sumber daya utama adalah alur. Anda dapat menggunakan konsol OpenSearch Layanan, AWS CLI, API OpenSearch Penyerapan, atau AWS SDK untuk menambah, mengelola, dan menghapus tanda dari alur.

Topik

- Izin diperlukan
- Cara menggunakan tanda (konsol)
- Cara menggunakan tanda (AWS CLI)

Izin diperlukan

OpenSearchPenyerapan menggunakan izin AWS Identity and Access Management Access Analyzer (IAM) berikut untuk menandai saluran pipa:

- `osis:TagResource`
- `osis:ListTagsForResource`
- `osis:UntagResource`

Untuk informasi lebih lanjut tentang setiap izin, lihat Tindakan, sumber daya, kunci kondisi untuk OpenSearch Penyerapan di Referensi Otorisasi Layanan.

Cara menggunakan tanda (konsol)

Konsol adalah cara paling mudah untuk menandai alur.

Untuk membuat tag

1. Masuk ke konsol OpenSearch Amazon di https://console.aws.amazon.com/aos/home.
2. Pilih Konsumsi di panel navigasi kiri.
3. Pilih alur yang ingin Anda tambahkan tanda dan buka tab Tag.
4. Pilih Kelola dan Tambahkan tag baru.
5. Masukkan kunci tanda dan nilai opsional.
6. Pilih Save (Simpan).

Untuk menghapus tag, ikuti langkah yang sama dan pilih Hapus di halaman Kelola tag.

Untuk informasi lebih lanjut tentang menggunakan konsol untuk bekerja dengan tanda, Panduan Memulai Konsol AWS Manajemen.

Cara menggunakan tanda (AWS CLI)

Untuk menandai pipeline menggunakanAWS CLI, kirim `TagResource` permintaan:

```
aws osis tag-resource
  --arn arn:aws:osis:us-east-1:123456789012:pipeline/my-pipeline
  --tags Key=service,Value=osis Key=source,Value=otel
```

Hapus tag dari pipeline menggunakan `UntagResource` perintah:

```
aws osis untag-resource
  --arn arn:aws:osis:us-east-1:123456789012:pipeline/my-pipeline
```

```
  --tag-keys service
```

Melihat tag yang ada untuk pipeline dengan `ListTagsForResource` perintah:

```
aws osis list-tags-for-resource
  --arn arn:aws:osis:us-east-1:123456789012:pipeline/my-pipeline
```

Pencatatan dan pemantauan Amazon OpenSearch Ingestion dengan Amazon CloudWatch

Amazon OpenSearch Ingestion menerbitkan metrik dan log ke Amazon. CloudWatch

Topik

- Memantau log
- Memantau metrik

Memantau log

Anda dapat mengaktifkan pencatatan untuk pipeline Amazon OpenSearch Ingestion untuk mengekspos pesan kesalahan dan peringatan yang dimunculkan selama operasi pipeline dan aktivitas konsumsi. OpenSearchPenyerapan menerbitkan semua log ke Amazon Logs. CloudWatch Log CloudWatch dapat memantau informasi dalam file log dan memberi tahu Anda ketika ambang tertentu terpenuhi. Anda juga dapat mengarsipkan data log Anda dalam penyimpanan yang sangat tahan lama. Untuk informasi selengkapnya, lihat Panduan Pengguna Log CloudWatch Amazon.

Log dari OpenSearch Penyerapan mungkin menunjukkan pemrosesan permintaan yang gagal, kesalahan otentikasi dari sumber ke wastafel, dan peringatan lain yang dapat membantu pemecahan masalah. Untuk log-nya, OpenSearch Tertelan menggunakan tingkat log`INFO`,, `WARN``ERROR`, dan. `FATAL` Sebaiknya aktifkan penerbitan log untuk semua jaringan pipa.

Izin diperlukan

Untuk mengaktifkan OpenSearch Penyerapan untuk mengirim log ke CloudWatch Log, Anda harus masuk sebagai pengguna yang memiliki izin IAM tertentu.

Anda memerlukan izin CloudWatch Log berikut untuk membuat dan memperbarui sumber daya pengiriman log:

```
{
    "Version": "2012-10-17",
    "Statement": [
        {
            "Effect": "Allow",
            "Resource": "*",
            "Action": [
                "logs:CreateLogDelivery",
                "logs:PutResourcePolicy",
                "logs:UpdateLogDelivery",
                "logs:DeleteLogDelivery",
                "logs:DescribeResourcePolicies",
                "logs:GetLogDelivery",
                "logs:ListLogDeliveries"
            ]
        }
    ]
}
```

Mengaktifkan penerbitan log

Anda dapat mengaktifkan penerbitan log pada pipeline yang ada, atau saat membuat pipeline. Untuk langkah-langkah untuk mengaktifkan penerbitan log selama pembuatan pipeline, lihatthe section called “Membuat jaringan pipa”.

Konsol

Untuk mengaktifkan penerbitan log pada pipeline yang sudah ada

1. Masuk ke konsol OpenSearch Layanan Amazon di https://console.aws.amazon.com/aos/home.
2. Pilih Penyerapan di panel navigasi kiri dan pilih pipeline yang ingin Anda aktifkan log.
3. Pilih Edit opsi penerbitan log.
4. Pilih Publikasikan ke CloudWatch Log.
5. Buat grup log baru atau pilih yang sudah ada. Kami menyarankan Anda memformat nama sebagai jalur, seperti`/aws/vendedlogs/OpenSearchIngestion/pipeline-name/audit-logs`. Format ini memudahkan penerapan kebijakan CloudWatch akses yang memberikan izin ke semua grup log di bawah jalur tertentu seperti. `/aws/vendedlogs/OpenSearchService/OpenSearchIngestion`

> **Important**
>
> Anda harus menyertakan awalan `vendedlogs` dalam nama grup log, jika tidak pembuatan gagal.

6. Pilih Save (Simpan).

CLI

Untuk mengaktifkan penerbitan log menggunakanAWS CLI, kirim permintaan berikut:

```
aws osis update-pipeline \
  --pipeline-name my-pipeline \
  --log-publishing-options  IsLoggingEnabled=true,CloudWatchLogDestination={LogGroup="/
aws/vendedlogs/OpenSearchIngestion/pipeline-name"}
```

Memantau metrik

Anda dapat memantau pipeline Amazon OpenSearch Ingestion menggunakan AmazonCloudWatch, yang mengumpulkan data mentah dan memprosesnya menjadi metrik yang dapat dibaca dan hampir waktu nyata. Statistik ini disimpan untuk jangka waktu 15 bulan, sehingga Anda dapat mengakses informasi historis dan mendapatkan perspektif yang lebih baik tentang performa aplikasi atau layanan web Anda. Anda juga dapat mengatur alarm yang memperhatikan ambang batas tertentu dan mengirim notifikasi atau mengambil tindakan saat ambang batas tersebut terpenuhi. Untuk informasi lebih lanjut, lihat Panduan Pengguna Amazon CloudWatch.

Konsol OpenSearch Ingestion menampilkan serangkaian grafik berdasarkan data mentah dari CloudWatch tab Kinerja untuk setiap pipeline.

OpenSearchPenyerapan melaporkan metrik dari sebagian besar plugin yang didukung. Jika plugin tertentu tidak memiliki tabel sendiri di bawah ini, itu berarti mereka tidak melaporkan metrik khusus plugin apa pun. Metrik pipeline dipublikasikan di `AWS/OSIS` namespace.

Topik

- Metrik umum
- Metrik buffer
- Metrik tanda tangan V4

- Metrik buffer pemblokiran terbatas
- Metrik sumber jejak Otel
- Metrik sumber metrik Otel
- Metrik http
- Metrik S3
- Metrik agregat
- Metrik tanggal
- Metrik Grok
- Metrik mentah Otel
- Metrik grup jejak Otel
- Metrik peta layanan
- Metrik OpenSearch
- Metrik sistem dan pengukuran

Metrik umum

Metrik berikut umum untuk semua prosesor dan sink.

Setiap metrik diawali dengan nama sub-pipeline dan nama plugin, dalam format < sub_pipeline_name >< plugin>< metric_name >. Misalnya, nama lengkap `recordsIn.count` metrik untuk sub-pipeline bernama `my-pipeline` dan prosesor tanggal akan`my-pipeline.date.recordsIn.count`.

sufiks	Deskripsi
`recordsIn.count`	Masuknya catatan ke komponen pipa. Metrik ini berlaku untuk prosesor dan sink. Statistik yang relevan: Jumlah Dimensi: `PipelineName`
`recordsOut.count`	Jalan keluar catatan dari komponen pipa. Metrik ini berlaku untuk prosesor dan sumber. Statistik yang relevan: Jumlah

sufiks	Deskripsi
	Dimensi: `PipelineName`
`timeElapsed.count`	Hitungan titik data yang direkam selama pelaksanaan komponen pipa. Metrik ini berlaku untuk prosesor dan sink. Statistik yang relevan: Jumlah Dimensi: `PipelineName`
`timeElapsed.sum`	Total waktu berlalu selama pelaksanaan komponen pipa. Metrik ini berlaku untuk prosesor dan tenggelam, dalam milidetik. Statistik yang relevan: Jumlah Dimensi: `PipelineName`
`timeElapsed.max`	Waktu maksimum berlalu selama pelaksanaan komponen pipa. Metrik ini berlaku untuk prosesor dan tenggelam, dalam milidetik. Statistik yang relevan: Maks Dimensi: `PipelineName`

Metrik buffer

Metrik berikut berlaku untuk buffer pemblokiran Bounded default yang secara otomatis dikonfigurasi oleh OpenSearch Ingestion untuk semua pipeline.

Setiap metrik diawali dengan nama sub-pipeline dan nama buffer, dalam format < sub_pipeline_name >< buffer_name >< metric_name >. Misalnya, nama lengkap `recordsWritten.count` metrik untuk sub-pipeline bernama `my-pipeline` akan`my-pipeline.BlockingBuffer.recordsWritten.count`.

sufiks	Deskripsi
`recordsWritten.count`	Jumlah catatan yang ditulis ke buffer.

sufiks	Deskripsi
	Statistik yang relevan: Jumlah Dimensi: `PipelineName`
`recordsRead.count`	Jumlah catatan dibaca dari buffer. Statistik yang relevan: Jumlah Dimensi: `PipelineName`
`recordsInFlight.value`	Jumlah catatan dicentang dibaca dari buffer. Statistik yang relevan: Rata-rata Dimensi: `PipelineName`
`recordsInBuffer.value`	Jumlah catatan saat ini dalam buffer. Statistik yang relevan: Rata-rata Dimensi: `PipelineName`
`recordsProcessed.count`	Jumlah catatan dibaca dari buffer dan diproses oleh pipa. Statistik yang relevan: Jumlah Dimensi: `PipelineName`
`recordsWriteFailed.count`	Jumlah catatan bahwa pipa gagal menulis ke wastafel. Statistik yang relevan: Jumlah Dimensi: `PipelineName`
`writeTimeElapsed.count`	Hitungan titik data yang direkam saat menulis ke buffer. Statistik yang relevan: Jumlah Dimensi: `PipelineName`

sufiks	Deskripsi
`writeTimeElapsed.sum`	Total waktu berlalu saat menulis ke buffer, dalam milidetik. Statistik yang relevan: Jumlah Dimensi: `PipelineName`
`writeTimeElapsed.max`	Waktu maksimum berlalu saat menulis ke buffer, dalam milidetik. Statistik yang relevan: Maks Dimensi: `PipelineName`
`writeTimeouts.count`	Hitungan menulis timeout ke buffer. Statistik yang relevan: Jumlah Dimensi: `PipelineName`
`readTimeElapsed.count`	Hitungan titik data yang direkam saat membaca dari buffer. Statistik yang relevan: Jumlah Dimensi: `PipelineName`
`readTimeElapsed.sum`	Total waktu berlalu saat membaca dari buffer, dalam milidetik. Statistik yang relevan: Jumlah Dimensi: `PipelineName`
`readTimeElapsed.max`	Waktu maksimum berlalu saat membaca dari buffer, dalam milidetik. Statistik yang relevan: Maks Dimensi: `PipelineName`

sufiks	Deskripsi
`checkpointTimeElapsed.count`	Hitungan poin data yang direkam saat checkpointing. Statistik yang relevan: Jumlah Dimensi: `PipelineName`
`checkpointTimeElapsed.sum`	Total waktu berlalu saat checkpointing, dalam milidetik. Statistik yang relevan: Jumlah Dimensi: `PipelineName`
`checkpointTimeElapsed.max`	Waktu maksimum berlalu saat checkpointing, dalam milidetik. Statistik yang relevan: Maks Dimensi: `PipelineName`

Metrik tanda tangan V4

Metrik berikut berlaku untuk titik akhir konsumsi untuk pipeline dan diasosiasikan dengan plugin sumber (`http`,, `otel_trace` dan). `otel_metrics` Semua permintaan ke endpoint konsumsi harus ditandatangani menggunakan Signature Version 4. Metrik ini dapat membantu Anda mengidentifikasi masalah otorisasi saat menyambung ke pipeline, atau mengonfirmasi bahwa Anda berhasil mengautentikasi.

Setiap metrik diawali dengan nama sub-pipeline dan. `osis_sigv4_auth` Sebagai contoh, *`sub_pipeline_name`*`.osis_sigv4_auth.httpAuthSuccess.count`.

sufiks	Deskripsi
`httpAuthSuccess.count`	Jumlah permintaan Signature V4 yang berhasil ke pipeline. Statistik yang relevan: Jumlah Dimensi: `PipelineName`

sufiks	Deskripsi
`httpAuthFailure.count`	Jumlah permintaan Signature V4 yang gagal ke pipeline. Statistik yang relevan: Jumlah Dimensi: `PipelineName`
`httpAuthServerError.count`	Jumlah permintaan Signature V4 ke pipeline yang mengembalikan kesalahan server. Statistik yang relevan: Jumlah Dimensi: `PipelineName`

Metrik buffer pemblokiran terbatas

Metrik berikut berlaku untuk buffer pemblokiran terbatas. Setiap metrik diawali dengan nama sub-pipeline dan. `BlockingBuffer` Sebagai contoh, *`sub_pipeline_name`*`.BlockingBuffer.bufferUsage.value`.

sufiks	Deskripsi
`bufferUsage.value`	Persen penggunaan `buffer_size` berdasarkan jumlah catatan dalam buffer. `buffer_size` mewakili jumlah maksimum catatan yang ditulis ke dalam buffer serta catatan dalam penerbangan yang belum diperiksa. Statistik yang relevan: Rata-rata Dimensi: `PipelineName`

Metrik sumber jejak Otel

Metrik berikut berlaku untuk sumber pelacakan Otel. Setiap metrik diawali dengan nama sub-pipeline dan. `otel_trace_source` Sebagai contoh, *`sub_pipeline_name`*`.otel_trace_source.requestTimeouts.count`.

sufiks	Deskripsi
`requestTimeouts.count`	Jumlah permintaan yang kehabisan waktu. Statistik yang relevan: Jumlah Dimensi: `PipelineName`
`requestsReceived.count`	Jumlah permintaan yang diterima oleh plugin. Statistik yang relevan: Jumlah Dimensi: `PipelineName`
`successRequests.count`	Jumlah permintaan yang berhasil diproses oleh plugin. Statistik yang relevan: Jumlah Dimensi: `PipelineName`
`badRequests.count`	Jumlah permintaan dengan format yang tidak valid yang diproses oleh plugin. Statistik yang relevan: Jumlah Dimensi: `PipelineName`
`requestsTooLarge.count`	Jumlah permintaan yang jumlah bentang dalam konten lebih besar dari kapasitas buffer. Statistik yang relevan: Jumlah Dimensi: `PipelineName`
`internalServerError.count`	Jumlah permintaan yang diproses oleh plugin dengan jenis pengecualian khusus. Statistik yang relevan: Jumlah Dimensi: `PipelineName`

sufiks	Deskripsi
`requestProcessDuration.count`	Hitungan titik data yang direkam saat memproses permintaan oleh plugin. Statistik yang relevan: Jumlah Dimensi: `PipelineName`
`requestProcessDuration.sum`	Latensi total permintaan yang diproses oleh plugin, dalam milidetik. Statistik yang relevan: Jumlah Dimensi: `PipelineName`
`requestProcessDuration.max`	Latensi maksimum permintaan yang diproses oleh plugin, dalam milidetik. Statistik yang relevan: Maks Dimensi: `PipelineName`
`payloadSize.count`	Hitungan distribusi ukuran payload permintaan masuk, dalam byte. Statistik yang relevan: Jumlah Dimensi: `PipelineName`
`payloadSize.sum`	Total distribusi ukuran payload permintaan masuk, dalam byte. Statistik yang relevan: Jumlah Dimensi: `PipelineName`

sufiks	Deskripsi
`payloadSize.max`	Distribusi maksimum ukuran muatan permintaan masuk, dalam byte. Statistik yang relevan: Maks Dimensi: `PipelineName`

Metrik sumber metrik Otel

Metrik berikut berlaku untuk sumber metrik Otel. Setiap metrik diawali dengan nama sub-pipeline dan. `otel_metrics_source` Sebagai contoh, *`sub_pipeline_name`*`.otel_metrics_source.requestTimeouts.count`.

sufiks	Deskripsi
`requestTimeouts.count`	Jumlah total permintaan untuk plugin yang kehabisan waktu. Statistik yang relevan: Jumlah Dimensi: `PipelineName`
`requestsReceived.count`	Jumlah total permintaan yang diterima oleh plugin. Statistik yang relevan: Jumlah Dimensi: `PipelineName`
`successRequests.count`	Jumlah permintaan berhasil diproses (200 kode status respon) oleh plugin. Statistik yang relevan: Jumlah Dimensi: `PipelineName`
`requestProcessDuration.count`	Hitungan latensi permintaan yang diproses oleh plugin, dalam hitungan detik.

sufiks	Deskripsi
	Statistik yang relevan: Jumlah Dimensi: `PipelineName`
`requestProcessDuration.sum`	Latensi total permintaan yang diproses oleh plugin, dalam milidetik. Statistik yang relevan: Jumlah Dimensi: `PipelineName`
`requestProcessDuration.max`	Latensi maksimum permintaan yang diproses oleh plugin, dalam milidetik. Statistik yang relevan: Maks Dimensi: `PipelineName`
`payloadSize.count`	Hitungan distribusi ukuran payload permintaan masuk, dalam byte. Statistik yang relevan: Jumlah Dimensi: `PipelineName`
`payloadSize.sum`	Total distribusi ukuran payload permintaan masuk, dalam byte. Statistik yang relevan: Jumlah Dimensi: `PipelineName`
`payloadSize.max`	Distribusi maksimum ukuran muatan permintaan masuk, dalam byte. Statistik yang relevan: Maks Dimensi: `PipelineName`

Metrik http

Metrik berikut berlaku untuk sumber HTTP. Setiap metrik diawali dengan nama sub-pipeline dan. `http` Sebagai contoh, *`sub_pipeline_name`*`.http.requestsReceived.count`.

sufiks	Deskripsi
`requestsReceived.count`	Jumlah permintaan yang diterima oleh `/log/ingest` titik akhir. Statistik yang relevan: Jumlah Dimensi: `PipelineName`
`requestsRejected.count`	Jumlah permintaan ditolak (429 kode status respon) oleh plugin. Statistik yang relevan: Jumlah Dimensi: `PipelineName`
`successRequests.count`	Jumlah permintaan berhasil diproses (200 kode status respon) oleh plugin. Statistik yang relevan: Jumlah Dimensi: `PipelineName`
`badRequests.count`	Jumlah permintaan dengan jenis konten yang tidak valid atau format (400 kode status respon) diproses oleh plugin. Statistik yang relevan: Jumlah Dimensi: `PipelineName`
`requestTimeouts.count`	Jumlah permintaan yang waktu di server sumber HTTP (415 kode status respon). Statistik yang relevan: Jumlah Dimensi: `PipelineName`

sufiks	Deskripsi
`requestsTooLarge.count`	Jumlah permintaan yang ukuran peristiwa dalam konten lebih besar dari kapasitas buffer (413 kode status respons). Statistik yang relevan: Jumlah Dimensi: `PipelineName`
`internalServerError.count`	Jumlah permintaan yang diproses oleh plugin dengan jenis pengecualian khusus (500 kode status respons). Statistik yang relevan: Jumlah Dimensi: `PipelineName`
`requestProcessDuration.count`	Hitungan latensi permintaan yang diproses oleh plugin, dalam hitungan detik. Statistik yang relevan: Jumlah Dimensi: `PipelineName`
`requestProcessDuration.sum`	Latensi total permintaan yang diproses oleh plugin, dalam milidetik. Statistik yang relevan: Jumlah Dimensi: `PipelineName`
`requestProcessDuration.max`	Latensi maksimum permintaan yang diproses oleh plugin, dalam milidetik. Statistik yang relevan: Maks Dimensi: `PipelineName`

sufiks	Deskripsi
`payloadSize.count`	Hitungan distribusi ukuran payload permintaan masuk, dalam byte. Statistik yang relevan: Jumlah Dimensi: `PipelineName`
`payloadSize.sum`	Total distribusi ukuran payload permintaan masuk, dalam byte. Statistik yang relevan: Jumlah Dimensi: `PipelineName`
`payloadSize.max`	Distribusi maksimum ukuran muatan permintaan masuk, dalam byte. Statistik yang relevan: Maks Dimensi: `PipelineName`

Metrik S3

Metrik berikut berlaku untuk sumber S3. Setiap metrik diawali dengan nama sub-pipeline dan. `s3` Sebagai contoh, *`sub_pipeline_name`*`.s3.s3ObjectsFailed.count`.

sufiks	Deskripsi
`s3ObjectsFailed.count`	Jumlah total objek S3 yang gagal dibaca plugin. Statistik yang relevan: Jumlah Dimensi: `PipelineName`
`s3ObjectsNotFound.count`	Jumlah objek S3 yang gagal dibaca plugin karena `Not Found` kesalahan dari S3. Metrik ini juga dihitung terhadap `s3ObjectsFailed` metrik.

sufiks	Deskripsi
	Statistik yang relevan: Jumlah Dimensi: `PipelineName`
`s3ObjectsAccessDenied.count`	Jumlah objek S3 yang gagal dibaca plugin karena `Forbidden` kesalahan `Access Denied` atau dari S3. Metrik ini juga dihitung terhadap `s3ObjectsFailed` metrik. Statistik yang relevan: Jumlah Dimensi: `PipelineName`
`s3ObjectReadTimeElapsed.count`	Jumlah waktu yang dibutuhkan plugin untuk melakukan permintaan GET untuk objek S3, mengurai, dan menulis peristiwa ke buffer. Statistik yang relevan: Jumlah Dimensi: `PipelineName`
`s3ObjectReadTimeElapsed.sum`	Jumlah total waktu yang diperlukan plugin untuk melakukan permintaan GET untuk objek S3, mengurai, dan menulis peristiwa ke buffer, dalam milidetik. Statistik yang relevan: Jumlah Dimensi: `PipelineName`
`s3ObjectReadTimeElapsed.max`	Jumlah maksimum waktu yang diperlukan plugin untuk melakukan permintaan GET untuk objek S3, mengurai, dan menulis peristiwa ke buffer, dalam milidetik. Statistik yang relevan: Maks Dimensi: `PipelineName`

sufiks	Deskripsi
`s3ObjectSizeBytes.count`	Hitungan distribusi ukuran objek S3, dalam byte. Statistik yang relevan: Jumlah Dimensi: `PipelineName`
`s3ObjectSizeBytes.sum`	Total distribusi ukuran objek S3, dalam byte. Statistik yang relevan: Jumlah Dimensi: `PipelineName`
`s3ObjectSizeBytes.max`	Distribusi maksimum ukuran objek S3, dalam byte. Statistik yang relevan: Maks Dimensi: `PipelineName`
`s3ObjectProcessedBytes.count`	Hitungan distribusi objek S3 diproses oleh plugin, dalam byte. Statistik yang relevan: Jumlah Dimensi: `PipelineName`
`s3ObjectProcessedBytes.sum`	Total distribusi objek S3 diproses oleh plugin, dalam byte. Statistik yang relevan: Jumlah Dimensi: `PipelineName`
`s3ObjectProcessedBytes.max`	Distribusi maksimum objek S3 diproses oleh plugin, dalam byte. Statistik yang relevan: Maks Dimensi: `PipelineName`

sufiks	Deskripsi
`s3ObjectsEvents.count`	Hitungan distribusi acara S3 yang diterima oleh plugin. Statistik yang relevan: Jumlah Dimensi: `PipelineName`
`s3ObjectsEvents.sum`	Total distribusi acara S3 yang diterima oleh plugin. Statistik yang relevan: Jumlah Dimensi: `PipelineName`
`s3ObjectsEvents.max`	Distribusi maksimum acara S3 yang diterima oleh plugin. Statistik yang relevan: Maks Dimensi: `PipelineName`
`sqsMessageDelay.count`	Hitungan titik data yang direkam sementara S3 mencatat waktu peristiwa untuk pembuatan objek ketika sepenuhnya diurai. Statistik yang relevan: Jumlah Dimensi: `PipelineName`
`sqsMessageDelay.sum`	Jumlah total waktu antara saat S3 merekam waktu peristiwa untuk pembuatan objek ketika sepenuhnya diurai, dalam milidetik. Statistik yang relevan: Jumlah Dimensi: `PipelineName`

sufiks	Deskripsi
`sqsMessageDelay.max`	Jumlah maksimum waktu antara saat S3 merekam waktu peristiwa untuk pembuatan objek ketika sepenuhnya diurai, dalam milidetik. Statistik yang relevan: Maks Dimensi: `PipelineName`
`s3ObjectsSucceeded.count`	Jumlah objek S3 yang berhasil dibaca plugin. Statistik yang relevan: Jumlah Dimensi: `PipelineName`
`sqsMessagesReceived.count`	Jumlah pesan Amazon SQS yang diterima dari antrian oleh plugin. Statistik yang relevan: Jumlah Dimensi: `PipelineName`
`sqsMessagesDeleted.count`	Jumlah pesan Amazon SQS dihapus dari antrian oleh plugin. Statistik yang relevan: Jumlah Dimensi: `PipelineName`
`sqsMessagesFailed.count`	Jumlah pesan Amazon SQS yang gagal diurai plugin. Statistik yang relevan: Jumlah Dimensi: `PipelineName`

Metrik agregat

Metrik berikut berlaku untuk prosesor Agregat. Setiap metrik diawali dengan nama sub-pipeline dan. `aggregate` Sebagai contoh, *`sub_pipeline_name`*`.aggregate.actionHandleEventsOut.count`.

sufiks	Deskripsi
`actionHandleEvents Out.count`	Jumlah peristiwa yang telah dikembalikan dari `handleEve nt` panggilan ke tindakan yang dikonfigurasi. Statistik yang relevan: Jumlah Dimensi: `PipelineName`
`actionHandleEvents Dropped.count`	Jumlah peristiwa yang telah dikembalikan dari `handleEve nt` panggilan ke tindakan yang dikonfigurasi. Statistik yang relevan: Jumlah Dimensi: `PipelineName`
`actionHandleEvents ProcessingErrors.count`	Jumlah panggilan yang dilakukan `handleEvent` untuk tindakan yang dikonfigurasi yang mengakibatkan kesalahan. Statistik yang relevan: Jumlah Dimensi: `PipelineName`
`actionConcludeGrou pEventsOut.count`	Jumlah peristiwa yang telah dikembalikan dari `concludeG roup` panggilan ke tindakan yang dikonfigurasi. Statistik yang relevan: Jumlah Dimensi: `PipelineName`
`actionConcludeGrou pEventsDropped.count`	Jumlah peristiwa yang belum dikembalikan dari `condludeGroup` panggilan ke tindakan yang dikonfigu rasi. Statistik yang relevan: Jumlah Dimensi: `PipelineName`

sufiks	Deskripsi
`actionConcludeGroupEventsProcessingErrors.count`	Jumlah panggilan yang dilakukan `concludeGroup` untuk tindakan yang dikonfigurasi yang mengakibatkan kesalahan. Statistik yang relevan: Jumlah Dimensi: `PipelineName`
`currentAggregateGroups.value`	Jumlah grup saat ini. Pengukur ini berkurang ketika kelompok disimpulkan, dan meningkat ketika suatu peristiwa memulai pembuatan grup baru. Statistik yang relevan: Rata-rata Dimensi: `PipelineName`

Metrik tanggal

Metrik berikut berlaku untuk prosesor Tanggal. Setiap metrik diawali dengan nama sub-pipeline dan. `date` Sebagai contoh, *`sub_pipeline_name`*`.date.dateProcessingMatchSuccess.count`.

sufiks	Deskripsi
`dateProcessingMatchSuccess.count`	Jumlah rekaman yang cocok setidaknya satu dari pola yang ditentukan dalam opsi `match` konfigurasi. Statistik yang relevan: Jumlah Dimensi: `PipelineName`
`dateProcessingMatchFailure.count`	Jumlah catatan yang tidak cocok dengan pola yang ditentukan dalam opsi `match` konfigurasi. Statistik yang relevan: Jumlah Dimensi: `PipelineName`

Metrik Grok

Metrik berikut berlaku untuk prosesor Grok. Setiap metrik diawali dengan nama sub-pipeline dan. `grok` Sebagai contoh, *`sub_pipeline_name`*`.grok.grokProcessingMatch.count`.

sufiks	Deskripsi
`grokProcessingMatch.count`	Jumlah catatan yang menemukan setidaknya satu pola cocok dari opsi `match` konfigurasi. Statistik yang relevan: Jumlah Dimensi: `PipelineName`
`grokProcessingMismatch.count`	Jumlah catatan yang tidak cocok dengan pola yang ditentukan dalam opsi `match` konfigurasi. Statistik yang relevan: Jumlah Dimensi: `PipelineName`
`grokProcessingErrors.count`	Jumlah kesalahan pemrosesan catatan. Statistik yang relevan: Jumlah Dimensi: `PipelineName`
`grokProcessingTimeouts.count`	Jumlah catatan yang habis waktu sementara pencocokan. Statistik yang relevan: Jumlah Dimensi: `PipelineName`
`grokProcessingTime.count`	Hitungan titik data yang direkam sementara catatan individu cocok dengan pola dari opsi `match` konfigurasi. Statistik yang relevan: Jumlah Dimensi: `PipelineName`

sufiks	Deskripsi
`grokProcessingTime.sum`	Jumlah total waktu yang dibutuhkan setiap rekaman individu untuk mencocokkan pola dari opsi `match` konfigurasi, dalam milidetik. Statistik yang relevan: Jumlah Dimensi: `PipelineName`
`grokProcessingTime.max`	Jumlah maksimum waktu yang dibutuhkan setiap rekaman individu untuk mencocokkan pola dari opsi `match` konfigurasi, dalam milidetik. Statistik yang relevan: Maks Dimensi: `PipelineName`

Metrik mentah Otel

Metrik berikut berlaku untuk prosesor mentah pelacakan Otel. Setiap metrik diawali dengan nama sub-pipeline dan. `otel_trace_raw` Sebagai contoh, *`sub_pipeline_name`*`.otel_trace_raw.traceGroupCacheCount.value`.

sufiks	Deskripsi
`traceGroupCacheCount.value`	Jumlah kelompok pelacakan dalam cache grup pelacakan. Statistik yang relevan: Jumlah Dimensi: `PipelineName`
`spanSetCount.value`	Jumlah rentang set dalam koleksi rentang set. Statistik yang relevan: Jumlah Dimensi: `PipelineName`

Metrik grup jejak Otel

Metrik berikut berlaku untuk prosesor grup pelacakan Otel. Setiap metrik diawali dengan nama sub-pipeline dan. `otel_trace_group` Sebagai contoh, *`sub_pipeline_name`*`.otel_trace_group.recordsInMissingTraceGroup.count`.

sufiks	Deskripsi
`recordsInMissingTraceGroup.count`	Jumlah rekaman masuknya hilang bidang kelompok jejak. Statistik yang relevan: Jumlah Dimensi: `PipelineName`
`recordsOutFixedTraceGroup.count`	Jumlah catatan jalan keluar dengan bidang kelompok jejak yang berhasil diisi. Statistik yang relevan: Jumlah Dimensi: `PipelineName`
`recordsOutMissingTraceGroup.count`	Jumlah catatan egress hilang bidang kelompok jejak. Statistik yang relevan: Jumlah Dimensi: `PipelineName`

Metrik peta layanan

Metrik berikut berlaku untuk prosesor stateful Service-map. Setiap metrik diawali dengan nama sub-pipeline dan. `service-map-stateful` Sebagai contoh, *`sub_pipeline_name`*`.service-map-stateful.spansDbSize.count`.

sufiks	Deskripsi
`spansDbSize.value`	Ukuran byte dalam memori dari bentang di MapDB di seluruh durasi jendela saat ini dan sebelumnya. Statistik yang relevan: Rata-rata

sufiks	Deskripsi
	Dimensi: `PipelineName`
`traceGroupDbSize.value`	Ukuran byte dalam memori dari kelompok jejak di MapDB di seluruh durasi jendela saat ini dan sebelumnya. Statistik yang relevan: Rata-rata Dimensi: `PipelineName`
`spansDbCount.value`	Hitungan bentang di MapDB di seluruh durasi jendela saat ini dan sebelumnya. Statistik yang relevan: Jumlah Dimensi: `PipelineName`
`traceGroupDbCount.value`	Hitungan kelompok jejak di MapDB di seluruh durasi jendela saat ini dan sebelumnya. Statistik yang relevan: Jumlah Dimensi: `PipelineName`
`relationshipCount.value`	Hitungan hubungan yang disimpan di seluruh durasi jendela saat ini dan sebelumnya. Statistik yang relevan: Jumlah Dimensi: `PipelineName`

Metrik OpenSearch

Metrik berikut berlaku untuk OpenSearchwastafel. Setiap metrik diawali dengan nama sub-pipeline dan. `opensearch` Sebagai contoh, *`sub_pipeline_name`*`.opensearch.bulkRequestErrors.count`.

sufiks	Deskripsi
`bulkRequestErrors.count`	Jumlah total kesalahan yang dihadapi saat mengirim permintaan massal. Statistik yang relevan: Jumlah Dimensi: `PipelineName`
`documentsSuccess.count`	Jumlah dokumen yang berhasil dikirim ke OpenSearch Layanan dengan permintaan massal, termasuk percobaan ulang. Statistik yang relevan: Jumlah Dimensi: `PipelineName`
`documentsSuccessFirstAttempt.count`	Jumlah dokumen berhasil dikirim ke OpenSearch Layanan dengan permintaan massal pada upaya pertama. Statistik yang relevan: Jumlah Dimensi: `PipelineName`
`documentErrors.count`	Jumlah dokumen yang gagal dikirim oleh permintaan massal. Statistik yang relevan: Jumlah Dimensi: `PipelineName`
`bulkRequestFailed.count`	Jumlah permintaan massal yang gagal. Statistik yang relevan: Jumlah Dimensi: `PipelineName`
`bulkRequestNumberOfRetries.count`	Jumlah retries dari permintaan massal yang gagal. Statistik yang relevan: Jumlah

sufiks	Deskripsi
	Dimensi: `PipelineName`
`bulkBadRequestErrors.count`	Jumlah `Bad Request` kesalahan yang dihadapi saat mengirim permintaan massal. Statistik yang relevan: Jumlah Dimensi: `PipelineName`
`bulkRequestNotAllowedErrors.count`	Jumlah `Request Not Allowed` kesalahan yang dihadapi saat mengirim permintaan massal. Statistik yang relevan: Jumlah Dimensi: `PipelineName`
`bulkRequestInvalidInputErrors.count`	Jumlah `Invalid Input` kesalahan yang dihadapi saat mengirim permintaan massal. Statistik yang relevan: Jumlah Dimensi: `PipelineName`
`bulkRequestNotFoundErrors.count`	Jumlah `Request Not Found` kesalahan yang dihadapi saat mengirim permintaan massal. Statistik yang relevan: Jumlah Dimensi: `PipelineName`
`bulkRequestTimeoutErrors.count`	Jumlah `Request Timeout` kesalahan yang dihadapi saat mengirim permintaan massal. Statistik yang relevan: Jumlah Dimensi: `PipelineName`

sufiks	Deskripsi
`bulkRequestServerErrors.count`	Jumlah `Server Error` kesalahan yang dihadapi saat mengirim permintaan massal. Statistik yang relevan: Jumlah Dimensi: `PipelineName`
`bulkRequestSizeBytes.count`	Hitungan distribusi ukuran payload permintaan massal, dalam byte. Statistik yang relevan: Jumlah Dimensi: `PipelineName`
`bulkRequestSizeBytes.sum`	Distribusi total ukuran payload permintaan massal, dalam byte. Statistik yang relevan: Jumlah Dimensi: `PipelineName`
`bulkRequestSizeBytes.max`	Distribusi maksimum ukuran payload permintaan massal, dalam byte. Statistik yang relevan: Maks Dimensi: `PipelineName`
`bulkRequestLatency.count`	Hitungan titik data yang direkam saat permintaan dikirim ke plugin, termasuk percobaan ulang. Statistik yang relevan: Jumlah Dimensi: `PipelineName`

sufiks	Deskripsi
`bulkRequestLatency.sum`	Latensi total permintaan yang dikirim ke plugin, termasuk percobaan ulang, dalam milidetik. Statistik yang relevan: Jumlah Dimensi: `PipelineName`
`bulkRequestLatency.max`	Latensi maksimum permintaan yang dikirim ke plugin, termasuk percobaan ulang, dalam milidetik. Statistik yang relevan: Maks Dimensi: `PipelineName`
`s3.dlqS3RecordsSuccess.count`	Jumlah catatan berhasil dikirim ke antrian surat mati S3. Statistik yang relevan: Jumlah Dimensi: `PipelineName`
`s3.dlqS3RecordsFailed.count`	Jumlah jalan yang gagal dikirim ke antrian surat mati S3. Statistik yang relevan: Jumlah Dimensi: `PipelineName`
`s3.dlqS3RequestSuccess.count`	Jumlah permintaan yang berhasil ke antrian surat mati S3. Statistik yang relevan: Jumlah Dimensi: `PipelineName`
`s3.dlqS3RequestFailed.count`	Jumlah permintaan gagal ke antrian surat mati S3. Statistik yang relevan: Jumlah Dimensi: `PipelineName`

sufiks	Deskripsi
`s3.dlqS3RequestLatency.count`	Hitungan titik data yang direkam saat permintaan dikirim ke antrian surat mati S3, termasuk percobaan ulang. Statistik yang relevan: Jumlah Dimensi: `PipelineName`
`s3.dlqS3RequestLatency.sum`	Latensi total permintaan yang dikirim ke antrian surat mati S3, termasuk percobaan ulang, dalam milidetik. Statistik yang relevan: Jumlah Dimensi: `PipelineName`
`s3.dlqS3RequestLatency.max`	Latensi maksimum permintaan yang dikirim ke antrian huruf mati S3, termasuk percobaan ulang, dalam milidetik. Statistik yang relevan: Maks Dimensi: `PipelineName`
`s3.dlqS3RequestSizeBytes.count`	Hitungan distribusi ukuran payload permintaan ke antrian surat mati S3, dalam byte. Statistik yang relevan: Jumlah Dimensi: `PipelineName`
`s3.dlqS3RequestSizeBytes.sum`	Distribusi total ukuran payload permintaan ke antrian surat mati S3, dalam byte. Statistik yang relevan: Jumlah Dimensi: `PipelineName`

sufiks	Deskripsi
`s3.dlqS3RequestSizeBytes.max`	Distribusi maksimum ukuran payload permintaan ke antrian surat mati S3, dalam byte. Statistik yang relevan: Maks Dimensi: `PipelineName`

Metrik sistem dan pengukuran

Metrik berikut berlaku untuk keseluruhan sistem OpenSearch Penyerapan. Metrik ini tidak diawali oleh apa pun.

Metrik	Deskripsi
`system.cpu.usage.value`	Persentase penggunaan CPU yang tersedia untuk semua node data. Statistik yang relevan: Rata-rata Dimensi:`PipelineName` ,`area`, `id`
`system.cpu.count.value`	Jumlah total penggunaan CPU untuk semua node data. Statistik yang relevan: Rata-rata Dimensi:`PipelineName` ,`area`, `id`
`jvm.memory.max.value`	Jumlah maksimum memori yang dapat digunakan untuk manajemen memori, dalam byte. Statistik yang relevan: Rata-rata Dimensi:`PipelineName` ,`area`, `id`
`jvm.memory.used.value`	Jumlah total memori yang digunakan, dalam byte. Statistik yang relevan: Rata-rata

Metrik	Deskripsi
	Dimensi:`PipelineName` ,`area`, `id` tanda
`jvm.memory.committed.value`	Jumlah memori yang berkomitmen untuk digunakan oleh mesin virtual Java (JVM), dalam byte. Statistik yang relevan: Rata-rata Dimensi:`PipelineName` ,`area`, `id`
`computeUnits`	Jumlah Unit OpenSearch Komputasi Tertelan (OCU Tertelan) yang digunakan oleh pipeline. Statistik yang relevan: Maks, Jumlah, Rata-rata Dimensi: `PipelineName`

Praktik terbaik untuk Amazon OpenSearch Ingestion

Topik ini memberikan praktik terbaik untuk membuat dan mengelola saluran Amazon OpenSearch Ingestion dan mencakup pedoman umum yang berlaku untuk banyak kasus penggunaan. Setiap beban kerja unik, dengan karakteristik unik, jadi tidak ada rekomendasi umum yang tepat untuk setiap kasus penggunaan.

Topik

Praktik terbaik umum

Praktik terbaik umum berikut berlaku untuk membuat dan mengelola jaringan pipa.

- Untuk memastikan ketersediaan tinggi, konfigurasikan pipeline VPC dengan dua atau tiga subnet. Jika Anda hanya menerapkan pipeline di satu subnet dan Availability Zone turun, Anda tidak akan dapat menyerap data.
- Dalam setiap pipa, kami sarankan untuk membatasi jumlah sub-pipeline menjadi 5 atau kurang.

- Jika Anda menggunakan plugin sumber S3, gunakan file S3 berukuran merata untuk kinerja optimal.
- Jika Anda menggunakan plugin sumber S3, tambahkan 30 detik batas waktu visibilitas tambahan untuk setiap ukuran file 0,25 GB di bucket S3 untuk kinerja optimal.
- Sertakan antrian surat mati (DLQ) dalam konfigurasi pipeline sehingga Anda dapat membongkar peristiwa yang gagal dan membuatnya dapat diakses untuk dianalisis. Jika sink Anda menolak data karena pemetaan yang salah atau masalah lain, Anda dapat merutekan data ke DLQ untuk memecahkan masalah dan memperbaiki masalah.

CloudWatch Alarm yang direkomendasikan

CloudWatch alarm melakukan tindakan ketika CloudWatch metrik melebihi nilai yang ditentukan untuk beberapa waktu. Misalnya, Anda mungkin ingin AWS untuk mengirim email kepada Anda jika status kesehatan klaster Anda `red` selama lebih dari satu menit. Bagian ini mencakup beberapa alarm yang direkomendasikan untuk Amazon OpenSearch Ingestion dan cara menanggapinya.

Untuk informasi selengkapnya tentang mengonfigurasi alarm, lihat Membuat CloudWatchAlarm Amazon di Panduan Pengguna Amazon CloudWatch .

Alarm	Isu
`computeUnits` maksimum adalah = dikonfigurasi `maxUnits` selama 15 menit, 3 kali berturut-turut	Pipa telah mencapai kapasitas maksimum dan mungkin memerlukan `maxUnits` pembaruan. Tingkatkan kapasitas maksimum pipa Anda
`opensearch.documentErrors.count` jumlah adalah = *{sub_pipeline_name}* `.opensearch.recordsIn.count` jumlah	Pipa tidak dapat menulis ke OpenSearch wastafel. Periksa izin pipeline dan konfirmasikan bahwa domain atau koleksinya sehat. Anda juga dapat memeriksa antrian huruf mati (DLQ) untuk peristiwa yang gagal, jika sudah dikonfigurasi.

Alarm	Isu
untuk 1 menit, 1 waktu berturut-turut	
`bulkRequestLatency.max` maks adalah> = x selama 1 menit, 1 waktu berturut-turut	Pipa mengalami latensi tinggi mengirim data ke OpenSearch wastafel. Ini kemungkinan karena wastafel berukuran terlalu kecil, atau strategi sharding yang buruk, yang menyebabkan wastafel tertinggal. Latensi tinggi yang berkelanjutan dapat memengaruhi kinerja pipa dan kemungkinan akan menyebabkan tekanan balik pada klien.
`httpAuthFailure.count` jumlah> = 1 selama 1 menit, 1 kali berturut-turut	Permintaan konsumsi tidak diautentikasi. Konfirmasikan bahwa semua klien memiliki otentikasi Signature Version 4 yang diaktifkan dengan benar.
`system.cpu.usage.value` rata-rata> = 80% selama 15 menit, 3 kali berturut-turut	Penggunaan CPU tinggi yang berkelanjutan bisa menjadi masalah. Pertimbangkan untuk meningkatkan kapasitas maksimum untuk pipa.
`bufferUsage.value` rata-rata > = 80% selama 15 menit, 3 kali berturut-turut	Penggunaan buffer tinggi yang berkelanjutan bisa menjadi masalah. Pertimbangkan untuk meningkatkan kapasitas maksimum untuk pipa.

Alarm lain yang mungkin Anda pertimbangkan

Pertimbangkan untuk mengonfigurasi alarm berikut tergantung pada fitur Amazon OpenSearch Ingestion yang biasa Anda gunakan.

Alarm	Isu
`dynamodb.exportJob`	Upaya untuk memicu ekspor ke Amazon S3 gagal.

Alarm	Isu
`Failure.c ount` jumlah 1	
`opensearc h.EndtoEn dLatency. avg` rata-rata> X selama 15 menit, 4 kali berturut-turut	`EndtoEndLatency` Lebih tinggi dari yang diinginkan untuk membaca dari aliran DynamoDB. Hal ini dapat disebabkan oleh OpenSearc h cluster underscaled atau kapasitas OCU pipeline maksimum yang terlalu rendah untuk throughput WCU pada tabel DynamoDB. `EndtoEndLatency` akan lebih tinggi setelah ekspor tetapi akan berkurang seiring waktu karena mengikuti aliran DynamoDB terbaru.
`dyanmodb. changeEve ntsProces sed.count` jumlah == 0 selama X menit	Tidak ada catatan yang dikumpulkan dari aliran DynamoDB. Ini bisa disebabkan oleh tidak adanya aktivitas di atas meja, atau masalah saat mengakses aliran DynamoDB.
`opensearc h.s3.dlqS 3RecordsS uccess.co unt` jumlah>= `opensearc h.documen tSuccess.count` jumlah selama 1 menit, 1 kali berturut-turut	Sejumlah besar catatan dikirim ke DLQ daripada wastafel. OpenSearch Tinjau metrik plugin OpenSearch sink untuk menyelidiki dan menentuka n akar penyebabnya.

Alarm	Isu
`grok.grok Processin gTimeouts .count` jumlah = recordsin.Hitung jumlah selama 1 menit, 5 kali berturut-turut	Semua data habis waktu sementara prosesor Grok mencoba mencocokkan pola. Ini kemungkinan memengaruhi kinerja dan memperlambat pipeline Anda. Pertimbangkan untuk menyesuaikan pola Anda untuk mengurangi batas waktu.
`grok.grok Processin gErrors.c ount` jumlah adalah >= 1 selama 1 menit, 1 waktu berturut-turut	Prosesor Grok gagal mencocokkan pola dengan data dalam pipeline, yang mengakibatkan kesalahan. Tinjau data Anda dan konfigurasi plugin Grok untuk memastikan pencocokan pola diharapkan.
`grok.grok Processin gMismatch .count` jumlah = recordsin.Hitung jumlah selama 1 menit, 5 kali berturut-turut	Prosesor Grok tidak dapat mencocokkan pola dengan data dalam pipeline. Tinjau data Anda dan konfigurasi plugin Grok untuk memastikan pencocokan pola diharapkan.
`date.date Processin gMatchFai lure.count` sum = recordsin.Hitung jumlah selama 1 menit, 5 kali berturut-turut	Prosesor Tanggal tidak dapat mencocokkan pola apa pun dengan data dalam pipeline. Tinjau data Anda dan konfigurasi plugin Tanggal untuk memastikan pola yang diharapkan.

Alarm	Isu
`s3.s3Objects Failed.count` jumlah> = 1 selama 1 menit, 1 kali berturut-turut	Masalah ini terjadi karena objek S3 tidak ada, atau pipeline memiliki hak istimewa yang tidak mencukupi. Reivew `s3ObjectsNotFound.count` dan `s3ObjectsAccessDenied.count` metrik untuk menentukan akar penyebabnya. Konfirmasikan bahwa objek S3 ada dan/atau perbarui izin.
`s3.sqsMessagesFailed.count` jumlah> = 1 selama 1 menit, 1 kali berturut-turut	Plugin S3 gagal memproses pesan Amazon SQS. Jika DLQ diaktifkan pada antrean SQS Anda, tinjau pesan yang gagal. Antrian mungkin menerima data tidak valid yang coba diproses oleh pipeline.
`http.badRequests.count` jumlah> = 1 selama 1 menit, 1 kali berturut-turut	Klien mengirim permintaan yang buruk. Konfirmasikan bahwa semua klien mengirimkan muatan yang tepat.
`http.requestsTooLarge.count` jumlah> = 1 selama 1 menit, 1 kali berturut-turut	Permintaan dari plugin sumber HTTP berisi terlalu banyak data, yang melebihi kapasitas buffer. Sesuaikan ukuran batch untuk klien Anda.
`http.internalServerError.count` jumlah> = 0 selama 1 menit, 1 kali berturut-turut	Plugin sumber HTTP mengalami kesulitan menerima acara.

Alarm	Isu
`http.requestTimeouts.count` jumlah> = 0 selama 1 menit, 1 kali berturut-turut	Batas waktu sumber kemungkinan merupakan hasil dari pipeline yang kurang tersedia. Pertimbangkan untuk meningkatkan pipa `maxUnits` untuk menangani beban kerja tambahan.
`otel_trace.badRequests.count` jumlah> = 1 selama 1 menit, 1 kali berturut-turut	Klien mengirim permintaan yang buruk. Konfirmasikan bahwa semua klien mengirimkan muatan yang tepat.
`otel_trace.requestsTooLarge.count` jumlah> = 1 selama 1 menit, 1 kali berturut-turut	Permintaan dari plugin sumber Otel Trace berisi terlalu banyak data, yang melebihi kapasitas buffer. Sesuaikan ukuran batch untuk klien Anda.
`otel_trace.internalServerError.count` jumlah> = 0 selama 1 menit, 1 kali berturut-turut	Plugin sumber Otel Trace mengalami kesulitan menerima acara.
`otel_trace.requestTimeouts.count` jumlah> = 0 selama 1 menit, 1 kali berturut-turut	Batas waktu sumber kemungkinan merupakan hasil dari pipeline yang kurang tersedia. Pertimbangkan untuk meningkatkan pipa `maxUnits` untuk menangani beban kerja tambahan.

Alarm	Isu
`otel_metrics.requestTimeouts.count` jumlah> = 0 selama 1 menit, 1 kali berturut-turut	Batas waktu sumber kemungkinan merupakan hasil dari pipeline yang kurang tersedia. Pertimbangkan untuk meningkatkan pipa `maxUnits` untuk menangani beban kerja tambahan.

Amazon Tanpa OpenSearch Server

Amazon OpenSearch Serverless adalah konfigurasi auto-scaling sesuai permintaan untuk Amazon Service. OpenSearch Koleksi OpenSearch Tanpa Server adalah OpenSearch klaster yang menskalakan kapasitas komputasi berdasarkan kebutuhan aplikasi Anda. Ini kontras dengan OpenSearch domain yang disediakan OpenSearch Layanan, yang Anda kelola kapasitasnya secara manual.

OpenSearch Tanpa server menyediakan opsi sederhana dan hemat biaya untuk beban kerja yang jarang, intermiten, atau tidak dapat diprediksi. Ini hemat biaya karena secara otomatis menskalakan kapasitas komputasi agar sesuai dengan penggunaan aplikasi Anda.

OpenSearch Koleksi tanpa server memiliki jenis volume penyimpanan berkapasitas tinggi, terdistribusi, dan sangat tersedia yang sama yang digunakan oleh domain Layanan yang disediakan OpenSearch .

OpenSearch Koleksi tanpa server selalu dienkripsi. Anda dapat memilih kunci enkripsi, tetapi tidak dapat menonaktifkan enkripsi. Untuk informasi selengkapnya, lihat the section called “Enkripsi”.

Topik

Manfaat

OpenSearch Tanpa server memiliki manfaat sebagai berikut:

- Lebih sederhana daripada yang disediakan - OpenSearch Tanpa server menghilangkan banyak kompleksitas pengelolaan OpenSearch cluster dan kapasitas. Ini secara otomatis mengukur dan menyetel cluster Anda, dan menangani manajemen siklus hidup shard dan indeks. Ini juga mengelola pembaruan perangkat lunak layanan dan peningkatan OpenSearch versi. Semua pembaruan dan peningkatan tidak mengganggu.
- Hemat biaya — Saat Anda menggunakan OpenSearch Tanpa Server, Anda hanya membayar sumber daya yang Anda konsumsi. Ini menghilangkan kebutuhan untuk penyediaan di muka dan penyediaan berlebihan untuk beban kerja puncak.
- Sangat tersedia - OpenSearch Tanpa server mendukung beban kerja produksi dengan redundansi untuk melindungi dari pemadaman Zona Ketersediaan dan kegagalan infrastruktur.
- Dapat Diskalakan — OpenSearch Tanpa server secara otomatis menskalakan sumber daya untuk mempertahankan tingkat konsumsi data dan waktu respons kueri yang cepat secara konsisten.

Apa itu Amazon Tanpa OpenSearch Server?

Amazon OpenSearch Serverless adalah konfigurasi tanpa server sesuai permintaan untuk Amazon Service. OpenSearch Tanpa server menghilangkan kompleksitas operasional penyediaan, konfigurasi, dan penyetelan cluster Anda. OpenSearch Ini adalah pilihan yang baik untuk organisasi yang tidak ingin mengelola sendiri OpenSearch cluster mereka, atau organisasi yang tidak memiliki sumber daya atau keahlian khusus untuk mengoperasikan cluster besar. Dengan OpenSearch Tanpa Server, Anda dapat dengan mudah mencari dan menganalisis volume data yang besar tanpa harus khawatir tentang infrastruktur dan manajemen data yang mendasarinya.

Koleksi OpenSearch tanpa server adalah sekelompok OpenSearch indeks yang bekerja sama untuk mendukung beban kerja atau kasus penggunaan tertentu. Koleksi lebih mudah digunakan daripada OpenSearch cluster yang dikelola sendiri, yang memerlukan penyediaan manual.

Koleksi memiliki jenis volume penyimpanan berkapasitas tinggi, terdistribusi, dan sangat tersedia yang sama yang digunakan oleh domain OpenSearch Layanan yang disediakan, tetapi mereka menghilangkan lebih banyak kerumitan karena tidak memerlukan konfigurasi dan penyetelan manual. Data dienkripsi dalam perjalanan dalam koleksi. OpenSearch Serverless juga mendukung OpenSearch Dasbor, yang menyediakan antarmuka intuitif untuk menganalisis data.

Koleksi tanpa server saat ini menjalankan OpenSearch versi 2.0.x. Saat versi baru dirilis, OpenSearch Tanpa Server akan secara otomatis meningkatkan koleksi Anda untuk menggunakan fitur baru, perbaikan bug, dan peningkatan kinerja.

Topik

Kasus penggunaan untuk Tanpa OpenSearch Server

OpenSearch Tanpa server mendukung dua kasus penggunaan utama:

- Analisis log - Segmen analisis log berfokus pada analisis volume besar data deret waktu semi-terstruktur yang dihasilkan mesin untuk wawasan operasional dan perilaku pengguna.
- Pencarian teks lengkap - Segmen pencarian teks lengkap mendukung aplikasi di jaringan internal Anda (sistem manajemen konten, dokumen hukum) dan aplikasi yang menghadap ke internet, seperti pencarian konten situs web e-niaga.

Saat Anda membuat koleksi, Anda memilih salah satu kasus penggunaan ini. Untuk informasi selengkapnya, lihat the section called “Memilih jenis koleksi”.

Memulai

Untuk memulai dengan OpenSearch Tanpa Server, buat satu atau beberapa koleksi menggunakan konsol OpenSearch Layanan AWS CLI, atau salah satu SDK. AWS Untuk tutorial yang membantu Anda mendapatkan koleksi dan berjalan dengan cepat, lihatthe section called “Memulai dengan Tanpa OpenSearch Server”.

OpenSearch Serverless mendukung operasi API ingest dan query yang sama dengan suite OpenSearch open source, sehingga Anda dapat terus menggunakan klien dan aplikasi yang ada. Klien Anda harus kompatibel dengan OpenSearch 2.x agar dapat bekerja dengan Tanpa OpenSearch Server. Untuk informasi selengkapnya, lihat the section called “Menelan data ke dalam koleksi”.

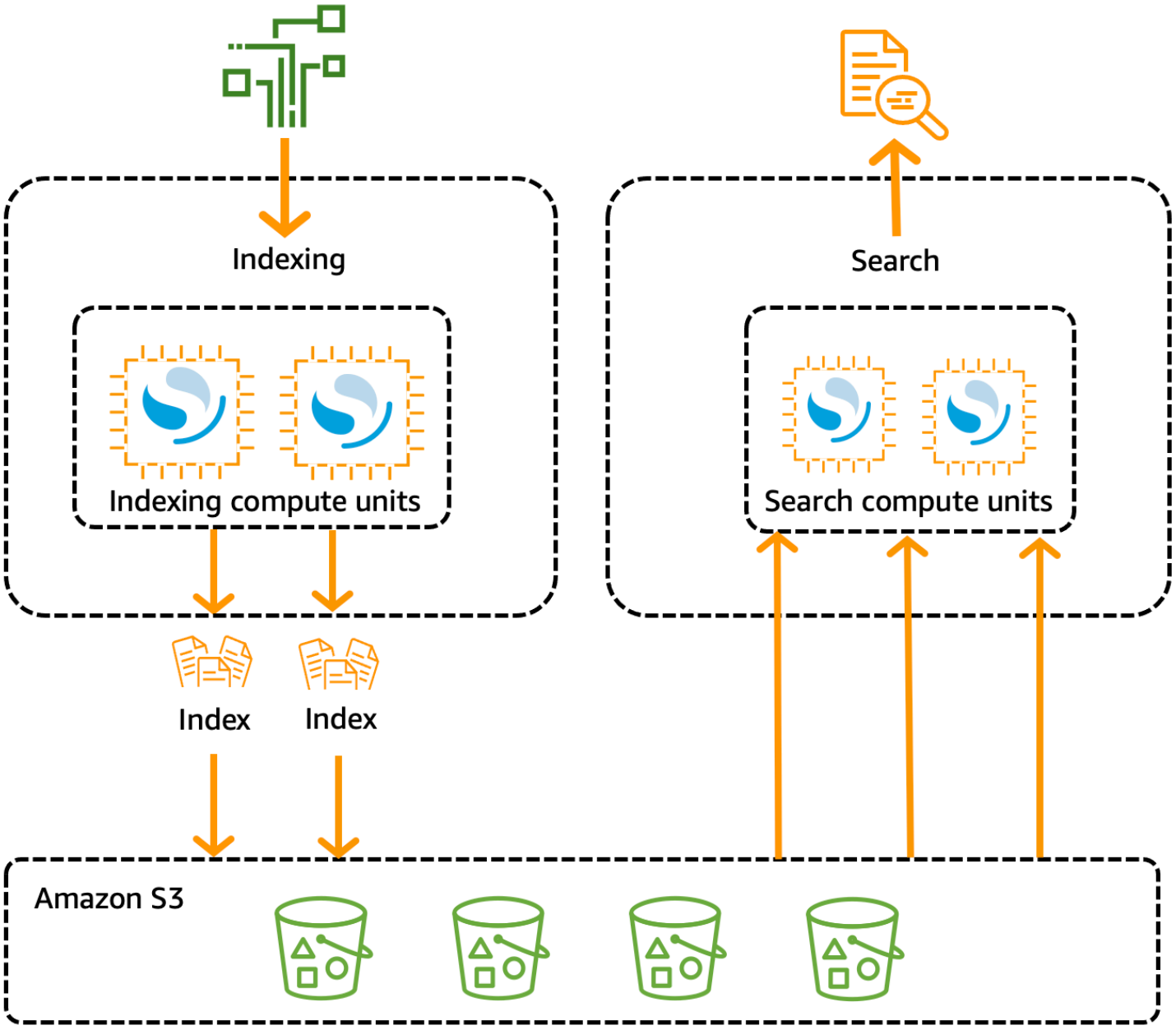
Cara kerjanya

OpenSearch Cluster tradisional memiliki satu set instance yang melakukan operasi pengindeksan dan pencarian, dan penyimpanan indeks digabungkan erat dengan kapasitas komputasi. Sebaliknya, OpenSearch Tanpa Server menggunakan arsitektur cloud-native yang memisahkan komponen pengindeksan (ingest) dari komponen penelusuran (kueri), dengan Amazon S3 sebagai penyimpanan data utama untuk indeks.

Arsitektur terpisah ini memungkinkan Anda menskalakan fungsi pencarian dan pengindeksan secara independen satu sama lain, dan secara independen dari data yang diindeks di S3. Arsitektur juga menyediakan isolasi untuk operasi ingest dan query sehingga mereka dapat berjalan secara bersamaan tanpa pertentangan sumber daya.

Saat Anda menulis data ke koleksi, OpenSearch Tanpa Server mendistribusikannya ke unit komputasi pengindeksan. Unit komputasi pengindeksan menyerap data yang masuk dan memindahkan indeks ke S3. Saat Anda melakukan penelusuran pada data pengumpulan, Rutekan OpenSearch Tanpa Server meminta ke unit komputasi penelusuran yang menyimpan data yang sedang ditanyakan. Unit komputasi pencarian mengunduh data yang diindeks langsung dari S3 (jika belum di-cache secara lokal), menjalankan operasi pencarian, dan melakukan agregasi.

Gambar berikut menggambarkan arsitektur terpisah ini:



OpenSearch Kapasitas komputasi tanpa server untuk konsumsi data, pencarian, dan kueri diukur dalam OpenSearch Compute Units (OCU). Setiap OCU adalah kombinasi dari 6 GiB memori dan CPU virtual yang sesuai (vCPU), serta transfer data ke Amazon S3. Setiap OCU mencakup penyimpanan singkat panas yang cukup untuk 120 GiB data indeks.

Saat Anda membuat koleksi pertama, OpenSearch Tanpa Server membuat instance dua OCUS —satu untuk pengindeksan dan satu untuk penelusuran. Untuk memastikan ketersediaan tinggi, ia juga meluncurkan satu set node siaga di Availability Zone lain. Untuk tujuan pengembangan dan pengujian, Anda dapat menonaktifkan pengaturan Aktifkan redundansi untuk koleksi, yang menghilangkan dua replika siaga dan hanya membuat instance dua OCU. Secara default, replika

aktif redundan diaktifkan, yang berarti bahwa total empat OCU dipakai untuk koleksi pertama dalam sebuah akun.

OCU ini ada bahkan ketika tidak ada aktivitas pada titik akhir koleksi apa pun. Semua koleksi berikutnya membagikan OCU ini. Saat Anda membuat koleksi tambahan di akun yang sama, OpenSearch Tanpa Server hanya menambahkan OCU tambahan untuk pencarian dan konsumsi sesuai kebutuhan untuk mendukung koleksi, sesuai dengan batas kapasitas yang Anda tentukan. Kapasitas turun kembali saat penggunaan komputasi Anda berkurang.

Untuk informasi tentang cara Anda ditagih untuk OCU ini, lihat. the section called “Harga untuk Tanpa OpenSearch Server”

Memilih jenis koleksi

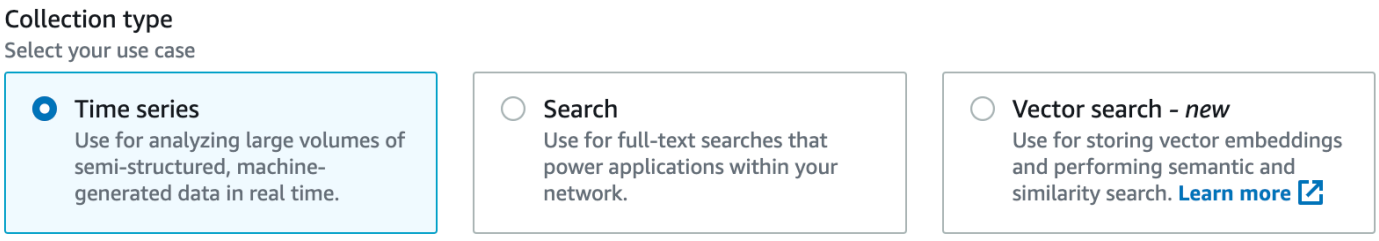
OpenSearch Serverless mendukung tiga jenis koleksi utama:

Time series — Segmen analisis log yang berfokus pada analisis volume besar data semi-terstruktur yang dihasilkan mesin secara real-time untuk operasional, keamanan, perilaku pengguna, dan wawasan bisnis.

Pencarian — Pencarian teks lengkap yang mendukung aplikasi di jaringan internal Anda (sistem manajemen konten, dokumen hukum) dan aplikasi yang menghadap ke internet, seperti pencarian situs web e-commerce dan pencarian konten.

Pencarian vektor — Pencarian semantik pada penyematan vektor yang menyederhanakan manajemen data vektor dan mendukung pengalaman pencarian tambahan pembelajaran mesin (ML) dan aplikasi AI generatif, seperti chatbots, asisten pribadi, dan deteksi penipuan.

Anda memilih jenis koleksi saat pertama kali membuat koleksi:



Jenis koleksi yang Anda pilih bergantung pada jenis data yang Anda rencanakan untuk dicerna ke dalam koleksi, dan bagaimana Anda berencana untuk menanyakannya. Anda tidak dapat mengubah jenis koleksi setelah Anda membuatnya.

Jenis koleksi memiliki perbedaan penting berikut:

- Untuk koleksi pencarian dan pencarian vektor, semua data disimpan dalam penyimpanan panas untuk memastikan waktu respons kueri yang cepat. Koleksi deret waktu menggunakan kombinasi penyimpanan panas dan hangat, di mana data terbaru disimpan dalam penyimpanan panas untuk mengoptimalkan waktu respons kueri untuk data yang lebih sering diakses.
- Untuk koleksi penelusuran deret waktu dan vektor, Anda tidak dapat mengindeks berdasarkan ID dokumen khusus atau memperbarui dengan permintaan upsert. Operasi ini dicadangkan untuk kasus penggunaan pencarian. Anda dapat memperbarui dengan ID dokumen sebagai gantinya. Untuk informasi selengkapnya, lihat the section called “Operasi dan izin OpenSearch API yang didukung”.
- Untuk koleksi penelusuran dan deret waktu, Anda tidak dapat menggunakan indeks tipe K-nN.

Harga untuk Tanpa OpenSearch Server

Di OpenSearch Tanpa Server, Anda dikenakan biaya untuk komponen berikut:

- Perhitungan konsumsi data
- Pencarian dan kueri komputasi
- Penyimpanan disimpan di Amazon S3

OCU ditagih setiap jam, dengan granularitas per detik. Dalam laporan akun Anda, Anda melihat entri untuk komputasi dalam OCU-jam dengan label untuk konsumsi data dan label untuk pencarian. Anda juga ditagih setiap bulan untuk data yang disimpan di Amazon S3. Anda tidak dikenakan biaya untuk menggunakan OpenSearch Dasbor.

Anda ditagih untuk minimal empat OCU yang dialokasikan untuk beban kerja Anda saat Anda membuat koleksi dan mengaktifkan replika aktif yang berlebihan. Anda ditagih minimal dua OCU untuk koleksi pertama di akun Anda jika Anda menonaktifkan replika aktif yang berlebihan. Semua koleksi berikutnya dapat membagikan OCU tersebut.

OpenSearch Tanpa server menambahkan OCU tambahan berdasarkan komputasi yang diperlukan untuk mendukung koleksi Anda. Jika beban kerja Anda menggunakan OCU fraksional, harganya proporsional. Anda dapat mengonfigurasi jumlah maksimum OCU untuk akun Anda untuk mengontrol biaya.

> **Note**
>
> Koleksi dengan unik tidak AWS KMS keys dapat berbagi OCU dengan koleksi lain.

OpenSearch Tanpa server mencoba menggunakan sumber daya minimum yang diperlukan untuk memperhitungkan perubahan beban kerja. Jumlah OCU yang disediakan pada waktu tertentu dapat bervariasi dan tidak tepat. Seiring waktu, algoritma yang digunakan OpenSearch Tanpa Server akan terus meningkat untuk meminimalkan penggunaan sistem dengan lebih baik.

Untuk detail harga selengkapnya, lihat harga OpenSearch Layanan Amazon.

Didukung Wilayah AWS

OpenSearch Tanpa server tersedia dalam subset OpenSearch Layanan Wilayah AWS yang tersedia di. Untuk daftar Wilayah yang didukung, lihat titik akhir dan kuota OpenSearch Layanan Amazon di. Referensi Umum AWS

Batasan

OpenSearch Tanpa server memiliki batasan sebagai berikut:

- Beberapa operasi OpenSearch API tidak didukung. Lihat the section called “Operasi dan izin OpenSearch API yang didukung”.
- Beberapa OpenSearch plugin tidak didukung. Lihat the section called “ OpenSearch Plugin yang didukung”.
- Saat ini tidak ada cara untuk memigrasikan data Anda secara otomatis dari domain OpenSearch Layanan terkelola ke koleksi tanpa server. Anda harus mengindeks ulang data Anda dari domain ke koleksi.
- Akses lintas akun ke koleksi tidak didukung. Anda tidak dapat menyertakan koleksi dari akun lain dalam enkripsi atau kebijakan akses data Anda.
- OpenSearch Plugin kustom tidak didukung.
- Anda tidak dapat mengambil atau memulihkan snapshot koleksi Tanpa OpenSearch Server.
- Pencarian dan replikasi Lintas Wilayah tidak didukung.
- Ada batasan jumlah sumber daya tanpa server yang dapat Anda miliki dalam satu akun dan Wilayah. Lihat OpenSearch Kuota tanpa server.

- Interval penyegaran untuk indeks dalam pencarian vektor dan koleksi deret waktu adalah sekitar 60 detik. Interval penyegaran untuk indeks dalam koleksi pencarian adalah sekitar 10 detik.
- Jumlah pecahan, jumlah interval, dan interval penyegaran tidak dapat dimodifikasi dan ditangani oleh Tanpa Server. OpenSearch Strategi sharding didasarkan pada jenis koleksi dan lalu lintas. Misalnya, pengumpulan deret waktu menskalakan pecahan primer berdasarkan kemacetan lalu lintas tulis.
- Fitur geospasial yang tersedia pada OpenSearch versi hingga 2.1 didukung.

Membandingkan OpenSearch Layanan dan Tanpa OpenSearch Server

Di OpenSearch Tanpa Server, beberapa konsep dan fitur berbeda dari fitur yang sesuai untuk domain Layanan yang disediakan OpenSearch . Misalnya, satu perbedaan penting adalah bahwa OpenSearch Serverless tidak memiliki konsep cluster atau node.

Tabel berikut menjelaskan bagaimana fitur dan konsep penting dalam OpenSearch Serverless berbeda dari fitur yang setara dalam domain Layanan yang disediakan OpenSearch .

Fitur	OpenSearch Layanan	OpenSearch Tanpa server
Domain versus koleksi	Indeks disimpan di domain, yang merupakan cluster yang telah disediakan sebelumnya OpenSearch . Untuk informasi selengkapnya, lihat Membuat dan mengelola domain.	Indeks disimpan dalam koleksi, yang merupakan pengelompokan logis indeks yang mewakili beban kerja atau kasus penggunaan tertentu. Untuk informasi selengkapnya, lihat the section called “Membuat, mencantumkan, dan menghapus koleksi”.
Jenis node dan manajemen kapasitas	Anda membangun cluster dengan tipe node yang memenuhi spesifikasi biaya dan kinerja Anda. Anda harus menghitung persyaratan penyimpanan Anda sendiri dan memilih jenis instans untuk domain Anda.	OpenSearch Tanpa server secara otomatis menskalakan dan menyediakan unit komputasi tambahan untuk akun Anda berdasarkan penggunaan kapasitas Anda. Untuk informasi selengkapnya, lihat the section called “Mengelola batas kapasitas”.

Fitur	OpenSearch Layanan	OpenSearch Tanpa server
	Untuk informasi selengkap nya, lihat the section called “Mengukur domain”.	
Penagihan	Anda membayar untuk setiap jam penggunaan instans EC2 dan untuk ukuran kumulatif volume penyimpanan EBS yang melekat pada instans Anda. Untuk informasi selengkapnya, lihat the section called “Harga untuk Amazon OpenSearch Service”.	Anda dikenakan biaya dalam OCU-jam untuk menghitung konsumsi data, menghitung penelusuran dan kueri, dan penyimpanan yang disimpan di S3. Untuk informasi selengkapnya, lihat the section called “Harga untuk Tanpa OpenSearch Server”.
Enkripsi	Enkripsi saat istirahat adalah opsional untuk domain. Untuk informasi selengkap nya, lihat the section called “Enkripsi diam”.	Enkripsi saat istirahat diperlukan untuk koleksi. Untuk informasi selengkapnya, lihat the section called “Enkripsi”.
Kontrol akses data	Akses ke data dalam domain ditentukan oleh kebijakan IAM dan kontrol akses berbutir halus.	Akses ke data dalam koleksi ditentukan oleh kebijakan akses data.
OpenSearc hOperasi yang didukung	OpenSearch Layanan mendukung subset dari semua operasi OpenSearch API. Untuk informasi selengkap nya, lihat the section called “Operasi yang didukung”.	OpenSearch Tanpa server mendukung subset operasi API yang berbeda. OpenSearch Untuk informasi selengkapnya, lihat the section called “Operasi dan plugin yang didukung”.

Fitur	OpenSearch Layanan	OpenSearch Tanpa server
Masuk dasbor	Masuk dengan nama pengguna dan kata sandi. Untuk informasi selengkap nya, lihat the section called “Mengakses OpenSearch Dasbor sebagai pengguna utama”.	Jika Anda masuk ke AWS konsol dan menavigasi ke URL Dasbor, Anda akan masuk secara otomatis. Untuk informasi selengkapnya, lihat the section called “Mengakses Dasbor OpenSearch ”.
API	Berinteraksi secara terprogram dengan OpenSearch Layanan menggunakan operasi API OpenSearch Layanan.	Berinteraksi secara terprogram dengan OpenSearch Tanpa Server menggunakan operasi API Tanpa Server. OpenSearch
Akses jaringan	Pengaturan jaringan untuk domain berlaku untuk titik akhir domain serta titik akhir OpenSearch Dasbor. Akses jaringan untuk keduanya digabungkan dengan erat.	Pengaturan jaringan untuk titik akhir domain dan titik akhir OpenSearch Dasbor dipisahka n. Anda dapat memilih untuk tidak mengonfig urasi akses jaringan untuk OpenSearch Dasbor. Untuk informasi selengkapnya, lihat the section called “Akses jaringan”.
Permintaan penandata nganan	Gunakan klien REST tingkat OpenSearch tinggi dan rendah untuk menandatangani permintaan. Tentukan nama layanan sebagaies.	Saat ini, OpenSearch Tanpa Server mendukung subset klien yang OpenSearch didukung Layanan. Saat Anda menandatangani permintaan, tentukan nama layanan sebagaiaoss. `x-amz-content-sha256` Header diperlukan. Untuk informasi selengkapnya, lihat the section called “Menandatangani permintaan HTTP dengan klien lain”.

Fitur	OpenSearch Layanan	OpenSearch Tanpa server
OpenSearch upgrade versi	Anda secara manual meng-upgrade domain Anda sebagai versi baru OpenSearch menjadi tersedia. Anda bertanggung jawab untuk memastikan bahwa domain Anda memenuhi persyarat an peningkatan, dan bahwa Anda telah mengatasi setiap perubahan yang melanggar.	OpenSearch Tanpa server secara otomatis meningkatkan koleksi Anda ke versi baru. OpenSearch Upgrade tidak selalu terjadi segera setelah versi baru tersedia.
Pembaruan perangkat lunak layanan	Anda secara manual menerapkan pembaruan perangkat lunak layanan ke domain Anda saat tersedia.	OpenSearch Tanpa server secara otomatis memperbarui koleksi Anda untuk menggunak an perbaikan bug, fitur, dan peningkatan kinerja terbaru.
Akses VPC	Anda dapat menyediakan domain Anda dalam VPC. Anda juga dapat membuat endpoint OpenSearch VPC yang dikelola Layanan tambahan untuk mengakses domain.	Anda membuat satu atau beberapa titik akhir VPC yang OpenSearch dikelola Tanpa Server untuk akun Anda. Kemudian, Anda menyertak an titik akhir ini dalam kebijakan jaringan.
Otentikasi SALL	Anda mengaktifkan otentikasi SAFL berdasarkan per domain. Untuk informasi selengkap nya, lihat the section called “Otentikasi SAMP untuk Dasbor OpenSearch ”.	Anda mengonfigurasi satu atau beberapa penyedia SAMP di tingkat akun, lalu Anda menyertakan ID pengguna dan grup terkait dalam kebijakan akses data. Untuk informasi selengkapnya, lihat the section called “Otentikasi SAMP”.

Fitur	OpenSearch Layanan	OpenSearch Tanpa server
Lapisan Keamanan Transportasi (TSL)	OpenSearch Layanan mendukung TLS 1.2 tetapi disarankan Anda menggunak an TLS 1.3.	OpenSearch Serverless mendukung TLS 1.2 tetapi disarankan Anda menggunakan TLS 1.3.

Memulai dengan Amazon Tanpa OpenSearch Server

Tutorial ini memandu Anda melalui langkah-langkah dasar untuk mendapatkan koleksi pencarian Amazon OpenSearch Tanpa Server dan berjalan dengan cepat. Koleksi pencarian memungkinkan Anda untuk memberi daya pada aplikasi di jaringan internal dan aplikasi yang menghadap ke internet, seperti pencarian situs web e-commerce dan pencarian konten.

Untuk mempelajari cara menggunakan koleksi pencarian vektor, lihatthe section called “Bekerja dengan koleksi pencarian vektor”. Untuk informasi lebih rinci tentang penggunaan koleksi, lihat the section called “Membuat, mencantumkan, dan menghapus koleksi” dan topik lainnya dalam panduan ini.

Anda akan menyelesaikan langkah-langkah berikut dalam tutorial ini:

1. Konfigurasikan izin
2. Buat koleksi
3. Unggah dan cari data
4. Hapus koleksi

Langkah 1: Konfigurasikan izin

Untuk menyelesaikan tutorial ini, dan untuk menggunakan OpenSearch Tanpa Server secara umum, Anda harus memiliki izin IAM yang benar. Dalam tutorial ini, Anda akan membuat koleksi, mengunggah dan mencari data, dan kemudian menghapus koleksi.

Pengguna atau peran Anda harus memiliki kebijakan berbasis identitas terlampir dengan izin minimum berikut:

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

```
  "Statement": [
    {
      "Action": [
        "aoss:CreateCollection",
        "aoss:ListCollections",
        "aoss:BatchGetCollection",
        "aoss:DeleteCollection",
        "aoss:CreateAccessPolicy",
        "aoss:ListAccessPolicies",
        "aoss:UpdateAccessPolicy",
        "aoss:CreateSecurityPolicy",
        "aoss:GetSecurityPolicy",
        "aoss:UpdateSecurityPolicy",
        "iam:ListUsers",
        "iam:ListRoles"
      ],
      "Effect": "Allow",
      "Resource": "*"
    }
  ]
}
```

Untuk informasi selengkapnya tentang izin IAM OpenSearch Tanpa Server, lihat. the section called “Pengelolaan Identitas dan Akses”

Langkah 2: Buat koleksi

Koleksi adalah sekelompok OpenSearch indeks yang bekerja sama untuk mendukung beban kerja atau kasus penggunaan tertentu.

Untuk membuat koleksi OpenSearch Tanpa Server

1. Buka konsol OpenSearch Layanan Amazon di https://console.aws.amazon.com/aos/home.
2. Pilih Koleksi di panel navigasi kiri dan pilih Buat koleksi.
3. Beri nama film koleksi.
4. Untuk jenis koleksi, pilih Cari. Untuk informasi selengkapnya, lihat Memilih jenis koleksi.
5. Untuk Keamanan, pilih Standard create.
6. Di bawah Enkripsi, pilih Gunakan Kunci milik AWS. Ini adalah AWS KMS key yang akan digunakan OpenSearch Tanpa Server untuk mengenkripsi data Anda.
7. Di bawah Jaringan, konfigurasikan pengaturan jaringan untuk koleksi.

- Untuk jenis akses, pilih Publik.
- Untuk jenis sumber daya, pilih Aktifkan akses ke OpenSearch titik akhir dan Aktifkan akses ke OpenSearch Dasbor. Karena Anda akan mengunggah dan mencari data menggunakan OpenSearch Dasbor, Anda harus mengaktifkan keduanya.

8. Pilih Berikutnya.
9. Untuk Mengonfigurasi akses data, atur pengaturan akses untuk koleksi. Kebijakan akses data memungkinkan pengguna dan peran untuk mengakses data dalam koleksi. Dalam tutorial ini, kami akan memberikan satu pengguna izin yang diperlukan untuk mengindeks dan mencari data dalam koleksi film.

   Buat aturan tunggal yang menyediakan akses ke koleksi film. Beri nama aturan Akses koleksi Film.
10. Pilih Tambahkan prinsipal, pengguna IAM, dan peran, lalu pilih pengguna atau peran yang akan Anda gunakan untuk masuk ke OpenSearch Dasbor dan mengindeks data. Pilih Simpan.
11. Di bawah Izin indeks, pilih semua izin.
12. Pilih Berikutnya.
13. Untuk pengaturan kebijakan akses, pilih Buat kebijakan akses data baru dan beri nama film kebijakan.
14. Pilih Berikutnya.
15. Tinjau pengaturan koleksi Anda dan pilih Kirim. Tunggu beberapa menit hingga status koleksi menjadi`Active`.

Langkah 3: Unggah dan cari data

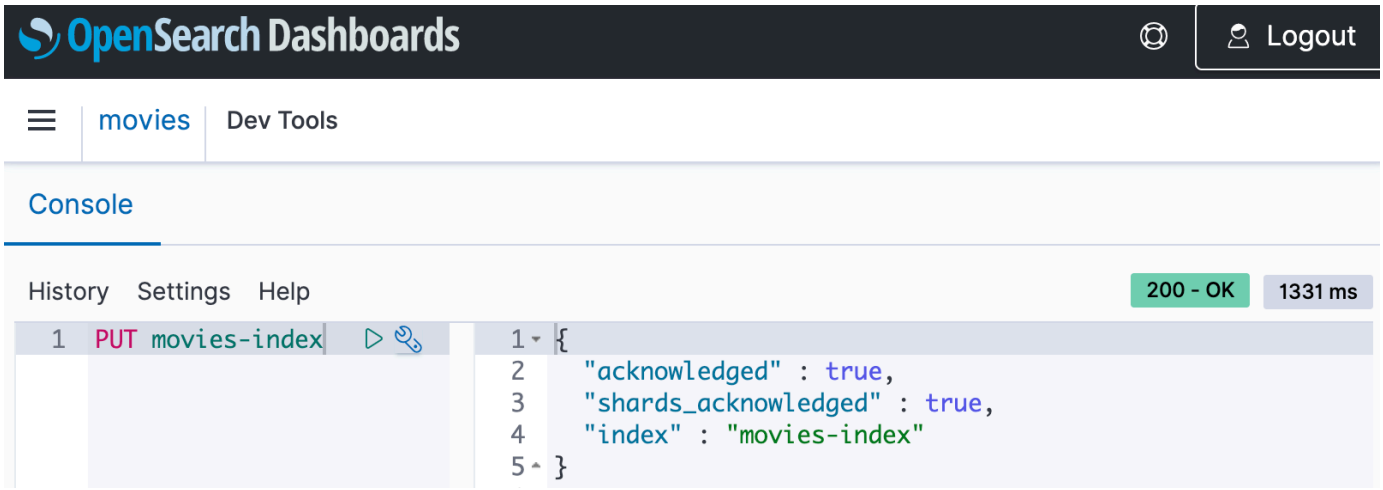
Anda dapat mengunggah data ke koleksi OpenSearch Tanpa Server menggunakan Postman atau cURL. Untuk singkatnya, contoh ini menggunakan Dev Tools dalam konsol OpenSearch Dashboards.

Untuk mengindeks dan mencari data dalam koleksi film

1. Pilih Koleksi di panel navigasi kiri dan pilih koleksi film untuk membuka halaman detailnya.
2. Pilih URL OpenSearch Dasbor untuk koleksi. URL mengambil format`https://dashboards.{region}.aoss.amazonaws.com/_login/?collectionId={collection-id}`.
3. Dalam OpenSearch Dashboards, buka panel navigasi kiri dan pilih Dev Tools.

4. Untuk membuat indeks tunggal yang disebut indeks film, kirim permintaan berikut:

```
PUT movies-index
```



5. Untuk mengindeks satu dokumen ke indeks film, kirim permintaan berikut:

```
PUT movies-index/_doc/1
{
  "title": "Shawshank Redemption",
  "genre": "Drama",
  "year": 1994
}
```

6. Untuk mencari data di OpenSearch Dasbor, Anda perlu mengonfigurasi setidaknya satu pola indeks. OpenSearch menggunakan pola-pola ini untuk mengidentifikasi indeks mana yang ingin Anda analisis. Buka panel navigasi kiri, pilih Stack Management, pilih Index Patterns, dan kemudian pilih Create index pattern. Untuk tutorial ini, masukkan film.
7. Pilih Langkah selanjutnya dan kemudian pilih Buat pola indeks. Setelah pola dibuat, Anda dapat melihat berbagai bidang dokumen seperti `title` dan `genre`.
8. Untuk mulai mencari data Anda, buka panel navigasi kiri lagi dan pilih Discover, atau gunakan API pencarian dalam Dev Tools.

Langkah 4: Hapus koleksi

Karena koleksi film adalah untuk tujuan pengujian, pastikan untuk menghapusnya ketika Anda selesai bereksperimen.

Untuk menghapus koleksi OpenSearch Tanpa Server

1. Kembali ke konsol OpenSearch Layanan Amazon.
2. Pilih Koleksi di panel navigasi kiri dan pilih koleksi film.
3. Pilih Hapus dan konfirmasi penghapusan.

Langkah selanjutnya

Sekarang setelah Anda tahu cara membuat koleksi dan indeks data, Anda mungkin ingin mencoba beberapa latihan berikut:

- Lihat opsi lanjutan lainnya untuk membuat koleksi. Untuk informasi selengkapnya, lihat the section called “Membuat, mencantumkan, dan menghapus koleksi”.
- Pelajari cara mengonfigurasi kebijakan keamanan untuk mengelola keamanan koleksi dalam skala besar. Untuk informasi selengkapnya, lihat the section called “Keamanan di Tanpa OpenSearch Server”.
- Temukan cara lain untuk mengindeks data ke dalam koleksi. Untuk informasi selengkapnya, lihat the section called “Menelan data ke dalam koleksi”.

Membuat dan mengelola koleksi Amazon OpenSearch Tanpa Server

Anda dapat membuat koleksi Amazon OpenSearch Tanpa Server menggunakan konsol, AWS CLI dan API, AWS SDK, dan. AWS CloudFormation

Topik

- Membuat, mencantumkan, dan menghapus koleksi Amazon Tanpa OpenSearch Server
- Bekerja dengan koleksi pencarian vektor
- Menggunakan kebijakan siklus hidup data dengan Amazon Serverless OpenSearch
- Menggunakan AWS SDK untuk berinteraksi dengan Amazon Tanpa Server OpenSearch
- Menggunakan AWS CloudFormation untuk membuat koleksi Amazon OpenSearch Tanpa Server

Membuat, mencantumkan, dan menghapus koleksi Amazon Tanpa OpenSearch Server

Koleksi di Amazon OpenSearch Tanpa Server adalah pengelompokan logis dari satu atau lebih indeks yang mewakili beban kerja analitik. OpenSearch Layanan secara otomatis mengelola dan menyetel koleksi, membutuhkan input manual minimal.

Topik

Izin diperlukan

OpenSearch Tanpa server menggunakan izin berikut AWS Identity and Access Management (IAM) untuk membuat dan mengelola koleksi. Anda dapat menentukan kondisi IAM untuk membatasi pengguna ke koleksi tertentu.

- `aoss:CreateCollection`— Buat koleksi.
- `aoss:ListCollections`— Daftar koleksi di akun saat ini.
- `aoss:BatchGetCollection`— Dapatkan detail tentang satu atau lebih koleksi.
- `aoss:UpdateCollection`— Memodifikasi koleksi.
- `aoss:DeleteCollection`— Hapus koleksi.

Contoh kebijakan akses berbasis identitas berikut memberikan izin minimum yang diperlukan bagi pengguna untuk mengelola satu koleksi bernama: `Logs`

```
[
   {
      "Sid":"Allows managing logs collections",
      "Effect":"Allow",
      "Action":[
         "aoss:CreateCollection",
```

```
            "aoss:ListCollections",
            "aoss:BatchGetCollection",
            "aoss:UpdateCollection",
            "aoss:DeleteCollection",
            "aoss:CreateAccessPolicy",
            "aoss:CreateSecurityPolicy"
         ],
         "Resource":"*",
         "Condition":{
            "StringEquals":{
               "aoss:collection":"Logs"
            }
         }
      }
]
```

`aoss:CreateAccessPolicy`dan `aoss:CreateSecurityPolicy` disertakan karena kebijakan enkripsi, jaringan, dan akses data diperlukan agar koleksi berfungsi dengan baik. Untuk informasi selengkapnya, lihat the section called “Pengelolaan Identitas dan Akses”.

> **Note**
>
> Jika Anda membuat koleksi pertama di akun Anda, Anda juga memerlukan `iam:CreateServiceLinkedRole` izin. Untuk informasi selengkapnya, lihat the section called “Peran pembuatan koleksi”.

Membuat koleksi

Anda dapat menggunakan konsol atau AWS CLI untuk membuat koleksi tanpa server. Langkah-langkah ini mencakup cara membuat koleksi pencarian atau deret waktu. Untuk membuat koleksi pencarian vektor, lihatthe section called “Bekerja dengan koleksi pencarian vektor”.

Buat koleksi (konsol)

Untuk membuat koleksi menggunakan konsol

1. Arahkan ke konsol OpenSearch Layanan Amazon di https://console.aws.amazon.com/aos/home/.
2. Perluas Tanpa Server di panel navigasi kiri dan pilih Koleksi.

3. Pilih Buat koleksi.
4. Berikan nama dan deskripsi untuk koleksi. Versi harus memenuhi kriteria berikut:

   - Ini unik untuk akun Anda dan Wilayah AWS
   - Dimulai dengan huruf kecil
   - Berisi antara 3 dan 32 karakter
   - Berisi hanya huruf kecil a-z, angka 0-9, dan tanda hubung (-)
5. Pilih jenis koleksi:

   - Pencarian — Pencarian teks lengkap yang mendukung aplikasi di jaringan internal Anda dan aplikasi yang menghadap ke internet. Semua data pencarian disimpan dalam penyimpanan panas untuk memastikan waktu respons kueri yang cepat.
   - Time series — Segmen analisis log yang berfokus pada analisis volume besar data semi-terstruktur yang dihasilkan mesin. Setidaknya 24 jam data disimpan pada indeks panas, dan sisanya tetap dalam penyimpanan hangat.
   - Pencarian vektor — Pencarian semantik pada embeddings vektor yang menyederhanakan manajemen data vektor. Memberdayakan pembelajaran mesin (ML) menambah pengalaman pencarian dan aplikasi AI generatif seperti chatbots, asisten pribadi, dan deteksi penipuan.

   Untuk informasi selengkapnya, lihat the section called “Memilih jenis koleksi”.
6. Di bawah Jenis Deployment, pilih pengaturan redundansi untuk koleksi Anda. Secara default, setiap koleksi dibuat dengan redundansi, yang berarti bahwa unit pengindeksan dan pencarian OpenSearch Compute Units (OCU) masing-masing memiliki replika siaga mereka sendiri di Availability Zone yang berbeda. Untuk tujuan pengembangan dan pengujian, Anda dapat memilih untuk menonaktifkan redundansi, yang mengurangi jumlah OCU dalam koleksi Anda menjadi dua. Untuk informasi selengkapnya, lihat the section called “Cara kerjanya”.
7. Di bawah Enkripsi, pilih AWS KMS kunci untuk mengenkripsi data Anda. OpenSearch Tanpa server memberi tahu Anda jika nama koleksi yang Anda masukkan cocok dengan pola yang ditentukan dalam kebijakan enkripsi. Anda dapat memilih untuk menyimpan kecocokan ini atau menggantinya dengan pengaturan enkripsi unik. Untuk informasi selengkapnya, lihat the section called “Enkripsi”.
8. Di bawah Pengaturan akses jaringan, konfigurasikan akses jaringan untuk koleksi.

   - Untuk jenis Akses, pilih publik atau pribadi. Kemudian, tentukan titik akhir VPC mana dan Layanan AWS dapat mengakses koleksi.

- Titik akhir VPC untuk akses — Tentukan satu atau beberapa titik akhir VPC untuk memungkinkan akses melalui. Untuk membuat titik akhir VPC, lihat. the section called “Titik akhir VPC”
- Layanan AWS akses pribadi — Pilih satu atau beberapa layanan yang didukung untuk memungkinkan akses ke.

- Untuk tipe Sumber Daya, pilih apakah koleksi akan dapat diakses melalui OpenSearchtitik akhir (untuk melakukan panggilan API melalui curl, Postman, dan sebagainya), melalui titik akhir OpenSearch Dasbor (untuk bekerja dengan visualisasi dan melakukan panggilan API melalui konsol), atau melalui keduanya.

> **Note**
>
> Layanan AWS akses pribadi hanya berlaku untuk OpenSearch titik akhir, bukan ke titik akhir OpenSearch Dasbor.

OpenSearch Tanpa server memberi tahu Anda jika nama koleksi yang Anda masukkan cocok dengan pola yang ditentukan dalam kebijakan jaringan. Anda dapat memilih untuk mempertahankan kecocokan ini atau menggantinya dengan pengaturan jaringan khusus. Untuk informasi selengkapnya, lihat the section called “Akses jaringan”.

9. (Opsional) Tambahkan satu atau lebih tag ke koleksi. Untuk informasi selengkapnya, lihat the section called “Penandaan koleksi”.
10. Pilih Berikutnya.
11. Konfigurasikan aturan akses data untuk koleksi, yang menentukan siapa yang dapat mengakses data dalam koleksi. Untuk setiap aturan yang Anda buat, lakukan langkah-langkah berikut:

    - Pilih Tambahkan prinsipal dan pilih satu atau beberapa peran IAM atau pengguna dan grup SAFL untuk menyediakan akses data.
    - Di bawah Berikan izin, pilih alias, templat, dan izin indeks untuk memberikan prinsip terkait. Untuk daftar lengkap izin dan akses yang mereka izinkan, lihatthe section called “Operasi dan izin OpenSearch API yang didukung”.

    OpenSearch Tanpa server memberi tahu Anda jika nama koleksi yang Anda masukkan cocok dengan pola yang ditentukan dalam kebijakan akses data. Anda dapat memilih untuk

mempertahankan kecocokan ini atau menggantinya dengan pengaturan akses data yang unik. Untuk informasi selengkapnya, lihat the section called “Kontrol akses data”.

12. Pilih Berikutnya.
13. Di bawah Pengaturan kebijakan akses data, pilih apa yang harus dilakukan dengan aturan yang baru saja Anda buat. Anda dapat menggunakannya untuk membuat kebijakan akses data baru, atau menambahkannya ke kebijakan yang ada.
14. Tinjau konfigurasi koleksi Anda dan pilih Kirim.

Status koleksi berubah menjadi `Creating` saat OpenSearch Tanpa Server membuat koleksi.

Buat koleksi (CLI)

Sebelum membuat koleksi menggunakan AWS CLI, Anda harus memiliki kebijakan enkripsi dengan pola sumber daya yang cocok dengan nama koleksi yang dimaksud. Misalnya, jika Anda berencana memberi nama aplikasi log koleksi, Anda dapat membuat kebijakan enkripsi seperti ini:

```
aws opensearchserverless create-security-policy \
  --name logs-policy \
  --type encryption --policy "{\"Rules\":[{\"ResourceType\":\"collection\",\"Resource
\":[\"collection\/logs-application\"]}],\"AWSOwnedKey\":true}"
```

Jika Anda berencana untuk menggunakan kebijakan untuk koleksi tambahan, Anda dapat membuat aturan lebih luas, seperti `collection/logs*` atau`collection/*`.

Anda juga perlu mengkonfigurasi pengaturan jaringan untuk koleksi dalam bentuk kebijakan jaringan. Menggunakan contoh aplikasi log sebelumnya, Anda dapat membuat kebijakan jaringan berikut:

```
aws opensearchserverless create-security-policy \
  --name logs-policy \
  --type network --policy "[{\"Description\":\"Public access for logs collection
\",\"Rules\":[{\"ResourceType\":\"dashboard\",\"Resource\":[\"collection\/logs-
application\"]},{\"ResourceType\":\"collection\",\"Resource\":[\"collection\/logs-
application\"]}],\"AllowFromPublic\":true}]"
```

> **Note**
>
> Anda dapat membuat kebijakan jaringan setelah membuat koleksi, tetapi sebaiknya lakukan terlebih dahulu.

Untuk membuat koleksi, kirim CreateCollectionpermintaan:

```
aws opensearchserverless create-collection --name "logs-application" --type SEARCH --
description "A collection for storing log data"
```

Untuk `type`, tentukan salah satu, `SEARCH` atau `TIMESERIES`. Untuk informasi selengkapnya, lihat the section called “Memilih jenis koleksi”.

Sampel respon

```
{
    "createCollectionDetail": {
        "id": "07tjusf2h91cunochc",
        "name": "books",
        "description":"A collection for storing log data",
        "status": "CREATING",
        "type": "SEARCH",
        "kmsKeyArn": "auto",
        "arn": "arn:aws:aoss:us-east-1:123456789012:collection/07tjusf2h91cunochc",
        "createdDate": 1665952577473
    }
}
```

Jika Anda tidak menentukan jenis koleksi dalam permintaan, defaultnya adalah. `TIMESERIES` Jika koleksi Anda dienkripsi dengan Kunci milik AWS, itu `kmsKeyArn` `auto` bukan ARN.

> **Important**
>
> Setelah membuat koleksi, Anda tidak akan dapat mengaksesnya kecuali cocok dengan kebijakan akses data. Untuk petunjuk untuk membuat kebijakan akses data, lihatthe section called “Kontrol akses data”.

Mengakses Dasbor OpenSearch

Setelah Anda membuat koleksi dengan AWS Management Console, Anda dapat menavigasi ke URL OpenSearch Dasbor koleksi. Anda dapat menemukan URL Dasbor dengan memilih Koleksi di panel navigasi kiri dan memilih koleksi untuk membuka halaman detailnya. URL mengambil format`https://dashboards.us-east-1.aoss.amazonaws.com/_login/?`

`collectionId=07tjusf2h91cunochc`. Setelah Anda menavigasi ke URL, Anda akan secara otomatis masuk ke Dasbor.

Jika Anda sudah memiliki URL OpenSearch Dasbor yang tersedia tetapi tidak ada AWS Management Console, memanggil URL Dasbor dari browser akan dialihkan ke konsol. Setelah Anda memasukkan AWS kredensil Anda, Anda akan secara otomatis masuk ke Dasbor. Untuk informasi tentang mengakses koleksi untuk SALL, lihat Mengakses OpenSearch Dasbor dengan SALL.

Batas waktu konsol OpenSearch Dasbor adalah satu jam dan tidak dapat dikonfigurasi.

> **Note**
>
> Pada 10 Mei 2023, OpenSearch memperkenalkan titik akhir global umum untuk OpenSearch Dasbor. Anda sekarang dapat menavigasi ke OpenSearch Dasbor di browser dengan URL yang mengambil formathttps://dashboards.*us-east-1*.aoss.amazonaws.com/_login/?collectionId=*07tjusf2h91cunochc*. Untuk memastikan kompatibilitas mundur, kami akan terus mendukung titik akhir OpenSearch Dasbor khusus koleksi yang ada dengan format. https://*07tjusf2h91cunochc*.*us-east-1*.aoss.amazonaws.com/_dashboards

Melihat koleksi

Anda dapat melihat koleksi yang ada Akun AWS di tab Koleksi di konsol OpenSearch Layanan Amazon.

Untuk membuat daftar koleksi beserta ID mereka, kirim ListCollectionspermintaan.

```
aws opensearchserverless list-collections
```

Sampel respon

```
{
   "collectionSummaries":[
      {
         "arn":"arn:aws:aoss:us-east-1:123456789012:collection/07tjusf2h91cunochc",
         "id":"07tjusf2h91cunochc",
         "name":"my-collection",
         "status":"CREATING"
```

```
      }
   ]
}
```

Untuk membatasi hasil pencarian, gunakan filter koleksi. Permintaan ini memfilter respons terhadap koleksi di `ACTIVE` negara bagian:

```
aws opensearchserverless list-collections --collection-filters '{ "status": "ACTIVE" }'
```

Untuk mendapatkan informasi lebih rinci tentang satu atau beberapa koleksi, termasuk OpenSearch titik akhir dan titik akhir OpenSearch Dasbor, kirim permintaan: BatchGetCollection

```
aws opensearchserverless batch-get-collection --ids ["07tjusf2h91cunochc",
 "1iu5usc4rame"]
```

> **Note**
>
> Anda dapat memasukkan `--names` atau `--ids` dalam permintaan, tetapi tidak keduanya.

Sampel respon

```
{
   "collectionDetails":[
      {
         "id": "07tjusf2h91cunochc",
         "name": "my-collection",
         "status": "ACTIVE",
         "type": "SEARCH",
         "description": "",
         "arn": "arn:aws:aoss:us-east-1:123456789012:collection/07tjusf2h91cunochc",
         "kmsKeyArn": "arn:aws:kms:us-
east-1:123456789012:key/1234abcd-12ab-34cd-56ef-1234567890ab",
         "createdDate": 1667446262828,
         "lastModifiedDate": 1667446300769,
         "collectionEndpoint": "https://07tjusf2h91cunochc.us-
east-1.aoss.amazonaws.com",
         "dashboardEndpoint": "https://07tjusf2h91cunochc.us-east-1.aoss.amazonaws.com/
_dashboards"
      },
```

```
      {
         "id": "178ukvtg3i82dvopdid",
         "name": "another-collection",
         "status": "ACTIVE",
         "type": "TIMESERIES",
         "description": "",
         "arn": "arn:aws:aoss:us-east-1:123456789012:collection/178ukvtg3i82dvopdid",
         "kmsKeyArn": "arn:aws:kms:us-
east-1:123456789012:key/1234abcd-12ab-34cd-56ef-1234567890ab",
         "createdDate": 1667446262828,
         "lastModifiedDate": 1667446300769,
         "collectionEndpoint": "https://178ukvtg3i82dvopdid.us-
east-1.aoss.amazonaws.com",
         "dashboardEndpoint": "https://178ukvtg3i82dvopdid.us-
east-1.aoss.amazonaws.com/_dashboards"
      }
   ],
   "collectionErrorDetails":[]
}
```

Menghapus koleksi

Menghapus koleksi akan menghapus semua data dan indeks dalam koleksi. Anda tidak dapat memulihkan koleksi setelah Anda menghapusnya.

Untuk menghapus koleksi menggunakan konsol

1. Dari panel Koleksi konsol OpenSearch Layanan Amazon, pilih koleksi yang ingin Anda hapus.
2. Pilih Hapus dan konfirmasikan penghapusan.

Untuk menghapus koleksi menggunakan AWS CLI, kirim DeleteCollectionpermintaan:

```
aws opensearchserverless delete-collection --id 07tjusf2h91cunochc
```

Sampel respon

```
{
   "deleteCollectionDetail":{
      "id":"07tjusf2h91cunochc",
      "name":"my-collection",
      "status":"DELETING"
```

```
    }
}
```

Bekerja dengan koleksi pencarian vektor

Jenis koleksi pencarian vektor di OpenSearch Tanpa Server memberikan kemampuan pencarian kesamaan yang dapat diskalakan dan berkinerja tinggi. Ini memudahkan Anda untuk membangun pengalaman pencarian tambahan pembelajaran mesin (ML) modern dan aplikasi kecerdasan buatan (AI) generatif tanpa harus mengelola infrastruktur basis data vektor yang mendasarinya.

Kasus penggunaan untuk koleksi pencarian vektor meliputi pencarian gambar, pencarian dokumen, pengambilan musik, rekomendasi produk, pencarian video, pencarian berbasis lokasi, deteksi penipuan, dan deteksi anomali.

Karena mesin vektor untuk OpenSearch Tanpa Server ditenagai oleh fitur pencarian k-nearest neighbor (k-NN) di OpenSearch, Anda mendapatkan fungsionalitas yang sama dengan kesederhanaan lingkungan tanpa server. Engine mendukung operasi OpenSearch API K-nn. Dengan operasi ini, Anda dapat memanfaatkan pencarian teks lengkap, pemfilteran lanjutan, agregasi, kueri geospasial, kueri bersarang untuk pengambilan data yang lebih cepat, dan hasil pencarian yang disempurnakan.

Mesin vektor menyediakan metrik jarak seperti jarak Euclidean, kesamaan kosinus, dan kesamaan produk titik, dan dapat menampung 16.000 dimensi. Anda dapat menyimpan bidang dengan berbagai tipe data untuk metadata, seperti angka, Boolean, tanggal, kata kunci, dan geopoint. Anda juga dapat menyimpan bidang dengan teks untuk informasi deskriptif untuk menambahkan lebih banyak konteks ke vektor yang disimpan. Colocating tipe data mengurangi kompleksitas, meningkatkan pemeliharaan, dan menghindari duplikasi data, tantangan kompatibilitas versi, dan masalah lisensi.

Memulai dengan koleksi pencarian vektor

Dalam tutorial ini, Anda menyelesaikan langkah-langkah berikut untuk menyimpan, mencari, dan mengambil embeddings vektor secara real time:

1. Konfigurasikan izin
2. Buat koleksi
3. Unggah dan cari data
4. Hapus koleksi

Langkah 1: Konfigurasikan izin

Untuk menyelesaikan tutorial ini (dan menggunakan OpenSearch Tanpa Server secara umum), Anda harus memiliki izin AWS Identity and Access Management (IAM) yang benar. Dalam tutorial ini, Anda membuat koleksi, mengunggah dan mencari data, dan kemudian menghapus koleksi.

Pengguna atau peran Anda harus memiliki kebijakan berbasis identitas terlampir dengan izin minimum berikut:

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Action": [
        "aoss:CreateCollection",
        "aoss:ListCollections",
        "aoss:BatchGetCollection",
        "aoss:DeleteCollection",
        "aoss:CreateAccessPolicy",
        "aoss:ListAccessPolicies",
        "aoss:UpdateAccessPolicy",
        "aoss:CreateSecurityPolicy",
        "iam:ListUsers",
        "iam:ListRoles"
      ],
      "Effect": "Allow",
      "Resource": "*"
    }
  ]
}
```

Untuk informasi selengkapnya tentang izin IAM OpenSearch Tanpa Server, lihat. the section called “Pengelolaan Identitas dan Akses”

Langkah 2: Buat koleksi

Koleksi adalah sekelompok OpenSearch indeks yang bekerja sama untuk mendukung beban kerja atau kasus penggunaan tertentu.

Untuk membuat koleksi OpenSearch Tanpa Server

1. Buka konsol OpenSearch Layanan Amazon di https://console.aws.amazon.com/aos/home.

2. Pilih Koleksi di panel navigasi kiri dan pilih Buat koleksi.
3. Beri nama perumahan koleksi.
4. Untuk jenis koleksi, pilih Pencarian vektor. Untuk informasi selengkapnya, lihat the section called "Memilih jenis koleksi".
5. Di bawah Jenis Deployment, hapus Aktifkan redundansi (replika aktif). Ini membuat koleksi dalam mode pengembangan atau pengujian, dan mengurangi jumlah Unit OpenSearch Komputasi (OCU) dalam koleksi Anda menjadi dua. Jika Anda ingin membuat lingkungan produksi dalam tutorial ini, biarkan kotak centang dipilih.
6. Di bawah Keamanan, pilih Mudah buat untuk merampingkan konfigurasi keamanan Anda. Semua data dalam mesin vektor dienkripsi dalam perjalanan dan diam secara default. Mesin vektor mendukung izin IAM berbutir halus sehingga Anda dapat menentukan siapa yang dapat membuat, memperbarui, dan menghapus enkripsi, jaringan, koleksi, dan indeks.
7. Pilih Berikutnya.
8. Tinjau pengaturan koleksi Anda dan pilih Kirim. Tunggu beberapa menit hingga status koleksi menjadi`Active`.

Langkah 3: Unggah dan cari data

Indeks adalah kumpulan dokumen dengan skema data umum yang menyediakan cara bagi Anda untuk menyimpan, mencari, dan mengambil embeddings vektor Anda dan bidang lainnya. Anda dapat membuat dan mengunggah data ke indeks dalam koleksi OpenSearch Tanpa Server dengan menggunakan konsol Alat Dev di OpenSearch Dasbor, atau alat HTTP seperti Postman atau awscurl. Tutorial ini menggunakan Dev Tools.

Untuk mengindeks dan mencari data dalam koleksi film

1. Untuk membuat indeks tunggal untuk koleksi baru Anda, kirim permintaan berikut di konsol Alat Dev. Secara default, ini menciptakan indeks dengan `nmslib` mesin dan jarak Euclidean.

```
PUT housing-index
{
   "settings": {
      "index.knn": true
   },
   "mappings": {
      "properties": {
         "housing-vector": {
            "type": "knn_vector",
```

```
            "dimension": 3
         },
         "title": {
            "type": "text"
         },
         "price": {
            "type": "long"
         },
         "location": {
            "type": "geo_point"
         }
      }
   }
}
```

2. Untuk mengindeks satu dokumen ke indeks perumahan, kirim permintaan berikut:

```
POST housing-index/_doc
{
  "housing-vector": [
    10,
    20,
    30
  ],
  "title": "2 bedroom in downtown Seattle",
  "price": "2800",
  "location": "47.71, 122.00"
}
```

3. Untuk mencari properti yang mirip dengan yang ada di indeks Anda, kirim kueri berikut:

```
GET housing-index/_search
{
    "size": 5,
    "query": {
        "knn": {
            "housing-vector": {
                "vector": [
                    10,
                    20,
                    30
                ],
                "k": 5
```

```
            }
        }
    }
}
```

Langkah 4: Hapus koleksi

Karena koleksi perumahan adalah untuk tujuan pengujian, pastikan untuk menghapusnya ketika Anda selesai bereksperimen.

Untuk menghapus koleksi OpenSearch Tanpa Server

1. Kembali ke konsol OpenSearch Layanan Amazon.
2. Pilih Koleksi di panel navigasi kiri dan pilih koleksi properti.
3. Pilih Hapus dan konfirmasikan penghapusan.

Pencarian yang difilter

Anda dapat menggunakan filter untuk menyempurnakan hasil pencarian semantik Anda. Untuk membuat indeks dan melakukan pencarian yang difilter pada dokumen Anda, ganti Upload dan cari data di tutorial sebelumnya dengan instruksi berikut. Langkah-langkah lainnya tetap sama. Untuk informasi selengkapnya tentang filter, lihat Penelusuran K-nn dengan filter.

Untuk mengindeks dan mencari data dalam koleksi film

1. Untuk membuat indeks tunggal untuk koleksi Anda, kirim permintaan berikut di konsol Alat Dev:

```
PUT housing-index-filtered
{
  "settings": {
    "index.knn": true
  },
  "mappings": {
    "properties": {
      "housing-vector": {
        "type": "knn_vector",
        "dimension": 3,
        "method": {
          "engine": "faiss",
          "name": "hnsw"
```

```
        }
      },
      "title": {
        "type": "text"
      },
      "price": {
        "type": "long"
      },
      "location": {
        "type": "geo_point"
      }
    }
  }
}
```

2. Untuk mengindeks satu dokumen ke dalam housing-index-filtered, kirim permintaan berikut:

```
POST housing-index-filtered/_doc
{
  "housing-vector": [
    10,
    20,
    30
  ],
  "title": "2 bedroom in downtown Seattle",
  "price": "2800",
  "location": "47.71, 122.00"
}
```

3. Untuk mencari data Anda untuk sebuah apartemen di Seattle dengan harga tertentu dan dalam jarak tertentu dari titik geografis, kirim permintaan berikut:

```
GET housing-index-filtered/_search
{
  "size": 5,
  "query": {
    "knn": {
      "housing-vector": {
        "vector": [
          0.1,
          0.2,
          0.3
        ],
```

```
        "k": 5,
        "filter": {
          "bool": {
            "must": [
              {
                "query_string": {
                  "query": "Find me 2 bedroom apartment in Seattle under $3000 ",
                  "fields": [
                    "title"
                  ]
                }
              },
              {
                "range": {
                  "price": {
                    "lte": 3000
                  }
                }
              },
              {
                "geo_distance": {
                  "distance": "100miles",
                  "location": {
                    "lat": 48,
                    "lon": 121
                  }
                }
              }
            ]
          }
        }
      }
    }
  }
}
```

Beban kerja skala miliar

Koleksi pencarian vektor mendukung beban kerja dengan miliaran vektor. Anda tidak perlu mengindeks ulang untuk tujuan penskalaan karena penskalaan otomatis melakukan ini untuk Anda. Jika Anda memiliki jutaan vektor (atau lebih) dengan jumlah dimensi yang tinggi dan membutuhkan

lebih dari 200 OCU, hubungi AWS Support untuk meningkatkan OpenSearch Compute Units (OCU) maksimum untuk akun Anda.

Batasan

Koleksi pencarian vektor memiliki batasan sebagai berikut:

- Koleksi pencarian vektor tidak mendukung mesin Apache Lucene ANN.
- Koleksi pencarian vektor hanya mendukung algoritma HNSW dengan Faiss dan tidak mendukung IVF dan IVFQ.
- Koleksi pencarian vektor tidak mendukung pemanasan, statistik, dan operasi API pelatihan model.
- Koleksi pencarian vektor tidak mendukung skrip sebaris atau tersimpan.
- Informasi jumlah indeks tidak tersedia di koleksi AWS Management Console pencarian vektor.
- Interval penyegaran untuk indeks pada koleksi pencarian vektor adalah 60 detik.

Langkah selanjutnya

Sekarang setelah Anda tahu cara membuat koleksi pencarian vektor dan data indeks, Anda mungkin ingin mencoba beberapa latihan berikut:

- Gunakan klien OpenSearch Python untuk bekerja dengan koleksi pencarian vektor. Lihat tutorial ini di GitHub.
- Gunakan klien OpenSearch Java untuk bekerja dengan koleksi pencarian vektor. Lihat tutorial ini di GitHub.
- Siapkan LangChain untuk digunakan OpenSearch sebagai penyimpanan vektor. LangChain adalah kerangka kerja open source untuk mengembangkan aplikasi yang didukung oleh model bahasa. Untuk informasi lebih lanjut, lihat dokumentasi LangChain .

Menggunakan kebijakan siklus hidup data dengan Amazon Serverless OpenSearch

Kebijakan siklus hidup data untuk pengumpulan deret waktu Amazon OpenSearch Tanpa Server menentukan masa pakai data dalam pengumpulan tersebut. OpenSearch Tanpa server menyimpan data selama periode waktu yang Anda konfigurasikan.

Anda dapat mengonfigurasi kebijakan siklus hidup data terpisah untuk setiap indeks dari setiap pengumpulan deret waktu di Anda. Akun AWS OpenSearch Tanpa server menyimpan dokumen dalam indeks untuk, setidaknya, periode penyimpanan yang Anda konfigurasikan dalam kebijakan. Kemudian secara otomatis menghapusnya dengan upaya terbaik, biasanya dalam 48 jam atau 10% dari periode retensi, mana yang lebih lama.

Hanya koleksi deret waktu yang mendukung kebijakan siklus hidup data. Mereka tidak didukung oleh pencarian atau koleksi pencarian vektor.

Topik

Kebijakan siklus hidup data

Dalam kebijakan siklus hidup data, Anda menentukan serangkaian aturan. Kebijakan siklus hidup data memungkinkan Anda mengelola periode penyimpanan data yang terkait dengan indeks atau koleksi yang cocok dengan aturan ini. Aturan-aturan ini menentukan periode retensi untuk data dalam indeks atau kelompok indeks. Setiap aturan terdiri dari tipe sumber daya (`index`), periode retensi, dan daftar sumber daya (indeks) yang berlaku untuk periode retensi.

Anda menentukan periode retensi dengan salah satu format berikut:

- `"MinIndexRetention": "24h"`— OpenSearch Tanpa server menyimpan data indeks untuk periode yang ditentukan dalam jam atau hari. Anda dapat mengatur periode ini menjadi dari `24h` ke`3650d`.
- `"NoMinIndexRetention": true`— OpenSearch Tanpa server mempertahankan data indeks tanpa batas waktu.

Dalam kebijakan sampel berikut, aturan pertama menetapkan periode retensi 15 hari untuk semua indeks dalam koleksi. `marketing` Aturan kedua menetapkan bahwa semua nama indeks yang dimulai dengan `log` `finance` koleksi tidak memiliki periode retensi yang ditetapkan dan akan dipertahankan tanpa batas waktu.

```
{
   "lifeCyclePolicyDetail": {
      "type": "retention",
      "name": "my-policy",
      "policyVersion": "MTY4ODI0NTM2OTk1N18x",
      "policy": {
         "Rules": [
            {
            "ResourceType":"index",
            "Resource":[
               "index/marketing/*"
            ],
            "MinIndexRetention": "15d"
         },
         {
            "ResourceType":"index",
            "Resource":[
               "index/finance/log*"
            ],
            "NoMinIndexRetention": true
         }
         ]
      },
      "createdDate": 1688245369957,
      "lastModifiedDate": 1688245369957
   }
}
```

Dalam aturan kebijakan contoh berikut, OpenSearch Tanpa Server menyimpan data di semua indeks untuk semua koleksi dalam akun tanpa batas waktu.

```
{
   "Rules": [
      {
         "ResourceType": "index",
         "Resource": [
            "index/*/*"
```

```
        ]
      }
   ],
   "NoMinIndexRetention": true
}
```

Izin diperlukan

Kebijakan siklus hidup untuk OpenSearch Tanpa Server menggunakan izin AWS Identity and Access Management (IAM) berikut. Anda dapat menentukan kondisi IAM untuk membatasi pengguna ke kebijakan siklus hidup data yang terkait dengan koleksi dan indeks tertentu.

- `aoss:CreateLifecyclePolicy`— Buat kebijakan siklus hidup data.
- `aoss:ListLifecyclePolicies`— Buat daftar semua kebijakan siklus hidup data di akun saat ini.
- `aoss:BatchGetLifecyclePolicy`— Melihat kebijakan siklus hidup data yang terkait dengan akun atau nama kebijakan.
- `aoss:BatchGetEffectiveLifecyclePolicy`— Melihat kebijakan siklus hidup data untuk sumber daya tertentu (`index`adalah satu-satunya sumber daya yang didukung).
- `aoss:UpdateLifecyclePolicy`— Ubah kebijakan siklus hidup data tertentu, dan ubah setelan retensi atau sumber dayanya.
- `aoss:DeleteLifecyclePolicy`— Hapus kebijakan siklus hidup data.

Kebijakan akses berbasis identitas berikut memungkinkan pengguna untuk melihat semua kebijakan siklus hidup data, dan memperbarui kebijakan dengan pola sumber daya: `collection/application-logs`

```
{
    "Version": "2012-10-17",
    "Statement": [
        {
            "Effect": "Allow",
            "Action": [
                "aoss:UpdateLifecyclePolicy"
            ],
            "Resource": "*",
            "Condition": {
                "StringEquals": {
```

```
                    "aoss:collection": "application-logs"
                }
            }
        },
        {
            "Effect": "Allow",
            "Action": [
                "aoss:ListLifecyclePolicies",
                "aoss:BatchGetLifecyclePolicy"
            ],
            "Resource": "*"
        }
    ]
}
```

Prioritas kebijakan

Mungkin ada situasi di mana aturan kebijakan siklus hidup data tumpang tindih, di dalam atau di seluruh kebijakan. Ketika ini terjadi, aturan dengan nama sumber daya yang lebih spesifik atau pola untuk indeks mengesampingkan aturan dengan nama sumber daya yang lebih umum atau pola untuk setiap indeks yang umum untuk kedua aturan.

Misalnya, dalam kebijakan berikut, dua aturan berlaku untuk indeks`index/sales/logstash`. Dalam situasi ini, aturan kedua diutamakan karena `index/sales/log*` merupakan pertandingan terpanjang. `index/sales/logstash` Oleh karena itu, OpenSearch Tanpa Server tidak menetapkan periode retensi untuk indeks.

```
{
      "Rules":[
         {
            "ResourceType":"index",
            "Resource":[
               "index/sales/*",
            ],
            "MinIndexRetention": "15d"
         },
         {
            "ResourceType":"index",
            "Resource":[
               "index/sales/log*",
            ],
            "NoMinIndexRetention": true
```

```
        }
      ]
  }
```

Sintaks kebijakan

Berikan satu atau lebih aturan. Aturan ini menentukan setelan siklus hidup data untuk indeks Tanpa OpenSearch Server Anda.

Setiap aturan berisi elemen-elemen berikut. Anda dapat memberikan `MinIndexRetention` atau `NoMinIndexRetention` dalam setiap aturan, tetapi tidak keduanya.

Elemen	Deskripsi
Jenis sumber daya	Jenis sumber daya yang berlaku untuk aturan tersebut. Satu-satunya opsi yang didukung untuk kebijakan siklus hidup data adalah. `index`
Sumber	Daftar nama dan/atau pola sumber daya. Pola terdiri dari awalan dan wildcard (*), yang memungkinkan izin terkait diterapka n ke beberapa sumber daya. Sebagai contoh, `index/`*`<collection-name|p attern>`* `/`*`<index-name|pattern>`* .
MinIndexRetention	Periode minimum, dalam hari (d) atau jam (h), untuk menyimpan dokumen dalam indeks. Batas bawah adalah `24h` dan batas atas adalah`3650d`.
NoMinIndexRetention	Jika`true`, OpenSearch Tanpa Server menyimpan dokumen tanpa batas waktu.

Berikut ini adalah beberapa contoh:

```
{
  "Rules": [
```

```
    {
      "ResourceType": "index",
      "Resource": [
        "index/autoparts-inventory/*"
      ],
      "MinIndexRetention": "20d"
    },
    {
      "ResourceType": "index",
      "Resource": [
        "index/auto*/gear"
      ],
      "MinIndexRetention": "24h"
    },
    {
      "ResourceType": "index",
      "Resource": [
        "index/autoparts-inventory/tires"
      ],
      "NoMinIndexRetention": true
    }
  ]
}
```

Membuat kebijakan siklus hidup data () AWS CLI

Untuk membuat kebijakan siklus hidup data menggunakan operasi API OpenSearch Tanpa Server, gunakan perintah. CreateLifecyclePolicy Perintah ini menerima kebijakan inline dan file.json. Kebijakan sebaris harus dikodekan sebagai string lolos JSON.

Permintaan berikut membuat kebijakan siklus hidup data:

```
aws opensearchserverless create-lifecycle-policy \
    --name my-policy \
    --type retention \
    --policy "{\"Rules\":[{\"ResourceType\":\"index\",\"Resource\":[\"index/autoparts-
inventory/*\"],\"MinIndexRetention\": \"81d\"},{\"ResourceType\":\"index\",\"Resource
\":[\"index/sales/orders*\"],\"NoMinIndexRetention\":true}]}"
```

Untuk menyediakan kebijakan dalam file JSON, gunakan format `--policy file://my-policy.json`

Melihat kebijakan siklus hidup data

Sebelum membuat koleksi, Anda mungkin ingin melihat pratinjau kebijakan siklus hidup data yang ada di akun Anda untuk melihat mana yang memiliki pola sumber daya yang cocok dengan nama koleksi Anda. ListLifecyclePoliciesPermintaan berikut mencantumkan semua kebijakan siklus hidup data di akun Anda:

```
aws opensearchserverless list-lifecycle-policies --type retention
```

Permintaan mengembalikan informasi tentang semua kebijakan siklus hidup data yang dikonfigurasi. Untuk melihat aturan pola yang ditentukan dalam satu kebijakan tertentu, cari informasi kebijakan dalam konten `lifecyclePolicySummaries` elemen dalam respons. Perhatikan `name` dan `type` kebijakan ini dan gunakan properti ini dalam BatchGetLifecyclePolicypermintaan untuk menerima tanggapan dengan rincian kebijakan berikut:

```
{
    "lifecyclePolicySummaries": [
        {
            "type": "retention",
            "name": "my-policy",
            "policyVersion": "MTY2MzY5MTY1MDA3Ml8x",
            "createdDate": 1663691650072,
            "lastModifiedDate": 1663691650072
        }
    ]
}
```

Untuk membatasi hasil pada kebijakan yang berisi koleksi atau indeks tertentu, Anda dapat menyertakan filter sumber daya:

```
aws opensearchserverless list-lifecycle-policies --type retention --resources
 "index/autoparts-inventory/*"
```

Untuk melihat informasi terperinci tentang kebijakan tertentu, gunakan BatchGetLifecyclePolicyperintah.

Memperbarui kebijakan siklus hidup data

Saat Anda mengubah kebijakan siklus hidup data, semua koleksi terkait akan terpengaruh. Untuk memperbarui kebijakan siklus hidup data di konsol OpenSearch Tanpa Server, perluas kebijakan siklus hidup data, pilih kebijakan yang akan diubah, lalu pilih Edit. Buat perubahan dan pilih Simpan.

Untuk memperbarui kebijakan siklus hidup data menggunakan API OpenSearch Tanpa Server, gunakan perintah. UpdateLifecyclePolicy Anda harus menyertakan versi kebijakan dalam permintaan. Anda dapat mengambil versi kebijakan dengan menggunakan `BatchGetLifecyclePolicy` perintah `ListLifecyclePolicies` atau. Menyertakan versi kebijakan terbaru memastikan bahwa Anda tidak secara tidak sengaja mengesampingkan perubahan yang dilakukan oleh orang lain.

Permintaan berikut memperbarui kebijakan siklus hidup data dengan dokumen JSON kebijakan baru:

```
aws opensearchserverless update-lifecycle-policy \
    --name my-policy \
    --type retention \
    --policy-version MTY2MzY5MTY1MDA3Ml8x \
    --policy file://my-new-policy.json
```

Mungkin ada jeda waktu beberapa menit antara saat Anda memperbarui kebijakan dan saat periode retensi baru diberlakukan.

Menghapus kebijakan siklus hidup data

Saat Anda menghapus kebijakan siklus hidup data, kebijakan tersebut tidak lagi berlaku untuk indeks yang cocok. Untuk menghapus kebijakan di konsol OpenSearch Tanpa Server, pilih kebijakan dan pilih Hapus.

Anda juga dapat menggunakan DeleteLifecyclePolicyperintah:

```
aws opensearchserverless delete-lifecycle-policy --name my-policy --type retention
```

Menggunakan AWS SDK untuk berinteraksi dengan Amazon Tanpa Server OpenSearch

Bagian ini mencakup contoh cara menggunakan AWS SDK untuk berinteraksi dengan Amazon Tanpa OpenSearch Server. Contoh kode ini menunjukkan cara membuat kebijakan dan koleksi keamanan, serta cara membuat kueri koleksi.

> **Note**
>
> Kami sedang membangun contoh kode ini. Jika Anda ingin menyumbangkan contoh kode (Java, Go, dll.), Silakan buka pull request langsung di dalam GitHubrepositori.

Topik

- Python
- JavaScript

Python

Contoh skrip berikut menggunakan AWS SDK for Python (Boto3), serta klien opensearch-py untuk Python, untuk membuat enkripsi, jaringan, dan kebijakan akses data, membuat koleksi yang cocok, dan mengindeks beberapa data sampel.

Untuk menginstal dependensi yang diperlukan, jalankan perintah berikut:

```
pip install opensearch-py
pip install boto3
pip install botocore
pip install requests-aws4auth
```

Di dalam skrip, ganti `Principal` elemen dengan Amazon Resource Name (ARN) pengguna atau peran yang menandatangani permintaan. Anda juga dapat memodifikasi file. `region`

```
from opensearchpy import OpenSearch, RequestsHttpConnection
from requests_aws4auth import AWS4Auth
import boto3
import botocore
import time

# Build the client using the default credential configuration.
# You can use the CLI and run 'aws configure' to set access key, secret
# key, and default region.

client = boto3.client('opensearchserverless')
service = 'aoss'
region = 'us-east-1'
credentials = boto3.Session().get_credentials()
```

```
awsauth = AWS4Auth(credentials.access_key, credentials.secret_key,
                   region, service, session_token=credentials.token)


def createEncryptionPolicy(client):
    """Creates an encryption policy that matches all collections beginning with tv-"""
    try:
        response = client.create_security_policy(
            description='Encryption policy for TV collections',
            name='tv-policy',
            policy="""
                {
                    \"Rules\":[
                        {
                            \"ResourceType\":\"collection\",
                            \"Resource\":[
                                \"collection\/tv-*\"
                            ]
                        }
                    ],
                    \"AWSOwnedKey\":true
                }
                """,
            type='encryption'
        )
        print('\nEncryption policy created:')
        print(response)
    except botocore.exceptions.ClientError as error:
        if error.response['Error']['Code'] == 'ConflictException':
            print(
                '[ConflictException] The policy name or rules conflict with an existing
 policy.')
        else:
            raise error


def createNetworkPolicy(client):
    """Creates a network policy that matches all collections beginning with tv-"""
    try:
        response = client.create_security_policy(
            description='Network policy for TV collections',
            name='tv-policy',
            policy="""
                [{
```

```
                    \"Description\":\"Public access for TV collection\",
                    \"Rules\":[
                        {
                            \"ResourceType\":\"dashboard\",
                            \"Resource\":[\"collection\/tv-*\"]
                        },
                        {
                            \"ResourceType\":\"collection\",
                            \"Resource\":[\"collection\/tv-*\"]
                        }
                    ],
                    \"AllowFromPublic\":true
                }]
                """,
            type='network'
        )
        print('\nNetwork policy created:')
        print(response)
    except botocore.exceptions.ClientError as error:
        if error.response['Error']['Code'] == 'ConflictException':
            print(
                '[ConflictException] A network policy with this name already exists.')
        else:
            raise error


def createAccessPolicy(client):
    """Creates a data access policy that matches all collections beginning with tv-"""
    try:
        response = client.create_access_policy(
            description='Data access policy for TV collections',
            name='tv-policy',
            policy="""
                [{
                    \"Rules\":[
                        {
                            \"Resource\":[
                                \"index\/tv-*\/*\"
                            ],
                            \"Permission\":[
                                \"aoss:CreateIndex\",
                                \"aoss:DeleteIndex\",
                                \"aoss:UpdateIndex\",
                                \"aoss:DescribeIndex\",
```

```
                            \"aoss:ReadDocument\",
                            \"aoss:WriteDocument\"
                        ],
                        \"ResourceType\": \"index\"
                    },
                    {
                        \"Resource\":[
                            \"collection\/tv-*\"
                        ],
                        \"Permission\":[
                            \"aoss:CreateCollectionItems\"
                        ],
                        \"ResourceType\": \"collection\"
                    }
                ],
                \"Principal\":[
                    \"arn:aws:iam::123456789012:role\/Admin\"
                ]
            }]
            """,
            type='data'
        )
        print('\nAccess policy created:')
        print(response)
    except botocore.exceptions.ClientError as error:
        if error.response['Error']['Code'] == 'ConflictException':
            print(
                '[ConflictException] An access policy with this name already exists.')
        else:
            raise error


def createCollection(client):
    """Creates a collection"""
    try:
        response = client.create_collection(
            name='tv-sitcoms',
            type='SEARCH'
        )
        return(response)
    except botocore.exceptions.ClientError as error:
        if error.response['Error']['Code'] == 'ConflictException':
            print(
```

```
                '[ConflictException] A collection with this name already exists. Try
 another name.')
        else:
            raise error


def waitForCollectionCreation(client):
    """Waits for the collection to become active"""
    response = client.batch_get_collection(
        names=['tv-sitcoms'])
    # Periodically check collection status
    while (response['collectionDetails'][0]['status']) == 'CREATING':
        print('Creating collection...')
        time.sleep(30)
        response = client.batch_get_collection(
            names=['tv-sitcoms'])
    print('\nCollection successfully created:')
    print(response["collectionDetails"])
    # Extract the collection endpoint from the response
    host = (response['collectionDetails'][0]['collectionEndpoint'])
    final_host = host.replace("https://", "")
    indexData(final_host)


def indexData(host):
    """Create an index and add some sample data"""
    # Build the OpenSearch client
    client = OpenSearch(
        hosts=[{'host': host, 'port': 443}],
        http_auth=awsauth,
        use_ssl=True,
        verify_certs=True,
        connection_class=RequestsHttpConnection,
        timeout=300
    )
    # It can take up to a minute for data access rules to be enforced
    time.sleep(45)

    # Create index
    response = client.indices.create('sitcoms-eighties')
    print('\nCreating index:')
    print(response)

    # Add a document to the index.
```

```
    response = client.index(
        index='sitcoms-eighties',
        body={
            'title': 'Seinfeld',
            'creator': 'Larry David',
            'year': 1989
        },
        id='1',
    )
    print('\nDocument added:')
    print(response)


def main():
    createEncryptionPolicy(client)
    createNetworkPolicy(client)
    createAccessPolicy(client)
    createCollection(client)
    waitForCollectionCreation(client)


if __name__ == "__main__":
    main()
```

JavaScript

Contoh skrip berikut menggunakan SDK untuk JavaScript di Node.js, serta klien opensearch-js untukJavaScript, untuk membuat enkripsi, jaringan, dan kebijakan akses data, membuat koleksi yang cocok, membuat indeks, dan mengindeks beberapa data sampel.

Untuk menginstal dependensi yang diperlukan, jalankan perintah berikut:

```
npm i aws-sdk
npm i aws4
npm i @opensearch-project/opensearch
```

Di dalam skrip, ganti `Principal` elemen dengan Amazon Resource Name (ARN) pengguna atau peran yang menandatangani permintaan. Anda juga dapat memodifikasi file. `region`

```
var AWS = require('aws-sdk');
var aws4 = require('aws4');
var {
```

```
    Client,
    Connection
} = require("@opensearch-project/opensearch");
var {
    OpenSearchServerlessClient,
    CreateSecurityPolicyCommand,
    CreateAccessPolicyCommand,
    CreateCollectionCommand,
    BatchGetCollectionCommand
} = require("@aws-sdk/client-opensearchserverless");
var client = new OpenSearchServerlessClient();

async function execute() {
    await createEncryptionPolicy(client)
    await createNetworkPolicy(client)
    await createAccessPolicy(client)
    await createCollection(client)
    await waitForCollectionCreation(client)
}

async function createEncryptionPolicy(client) {
    // Creates an encryption policy that matches all collections beginning with 'tv-'
    try {
        var command = new CreateSecurityPolicyCommand({
            description: 'Encryption policy for TV collections',
            name: 'tv-policy',
            type: 'encryption',
            policy: " \
        { \
            \"Rules\":[ \
                { \
                    \"ResourceType\":\"collection\", \
                    \"Resource\":[ \
                        \"collection\/tv-*\" \
                    ] \
                } \
            ], \
            \"AWSOwnedKey\":true \
        }"
        });
        const response = await client.send(command);
        console.log("Encryption policy created:");
        console.log(response['securityPolicyDetail']);
    } catch (error) {
```

```
        if (error.name === 'ConflictException') {
            console.log('[ConflictException] The policy name or rules conflict with an
 existing policy.');
        } else
            console.error(error);
    };
}

async function createNetworkPolicy(client) {
    // Creates a network policy that matches all collections beginning with 'tv-'
    try {
        var command = new CreateSecurityPolicyCommand({
            description: 'Network policy for TV collections',
            name: 'tv-policy',
            type: 'network',
            policy: " \
            [{ \
                \"Description\":\"Public access for television collection\", \
                \"Rules\":[ \
                    { \
                        \"ResourceType\":\"dashboard\", \
                        \"Resource\":[\"collection\/tv-*\"] \
                    }, \
                    { \
                        \"ResourceType\":\"collection\", \
                        \"Resource\":[\"collection\/tv-*\"] \
                    } \
                ], \
                \"AllowFromPublic\":true \
            }]"
        });
        const response = await client.send(command);
        console.log("Network policy created:");
        console.log(response['securityPolicyDetail']);
    } catch (error) {
        if (error.name === 'ConflictException') {
            console.log('[ConflictException] A network policy with that name already
 exists.');
        } else
            console.error(error);
    };
}

async function createAccessPolicy(client) {
```

```
    // Creates a data access policy that matches all collections beginning with 'tv-'
    try {
        var command = new CreateAccessPolicyCommand({
            description: 'Data access policy for TV collections',
            name: 'tv-policy',
            type: 'data',
            policy: " \
            [{ \
                \"Rules\":[ \
                    { \
                        \"Resource\":[ \
                            \"index\/tv-*\/*\" \
                        ], \
                        \"Permission\":[ \
                            \"aoss:CreateIndex\", \
                            \"aoss:DeleteIndex\", \
                            \"aoss:UpdateIndex\", \
                            \"aoss:DescribeIndex\", \
                            \"aoss:ReadDocument\", \
                            \"aoss:WriteDocument\" \
                        ], \
                        \"ResourceType\": \"index\" \
                    }, \
                    { \
                        \"Resource\":[ \
                            \"collection\/tv-*\" \
                        ], \
                        \"Permission\":[ \
                            \"aoss:CreateCollectionItems\" \
                        ], \
                        \"ResourceType\": \"collection\" \
                    } \
                ], \
                \"Principal\":[ \
                    \"arn:aws:iam::123456789012:role\/Admin\" \
                ] \
            }]"
        });
        const response = await client.send(command);
        console.log("Access policy created:");
        console.log(response['accessPolicyDetail']);
    } catch (error) {
        if (error.name === 'ConflictException') {
```

```
            console.log('[ConflictException] An access policy with that name already
 exists.');
        } else
            console.error(error);
    };
}

async function createCollection(client) {
    // Creates a collection to hold TV sitcoms indexes
    try {
        var command = new CreateCollectionCommand({
            name: 'tv-sitcoms',
            type: 'SEARCH'
        });
        const response = await client.send(command);
        return (response)
    } catch (error) {
        if (error.name === 'ConflictException') {
            console.log('[ConflictException] A collection with this name already
 exists. Try another name.');
        } else
            console.error(error);
    };
}

async function waitForCollectionCreation(client) {
    // Waits for the collection to become active
    try {
        var command = new BatchGetCollectionCommand({
            names: ['tv-sitcoms']
        });
        var response = await client.send(command);
        while (response.collectionDetails[0]['status'] == 'CREATING') {
            console.log('Creating collection...')
            await sleep(30000) // Wait for 30 seconds, then check the status again
            function sleep(ms) {
                return new Promise((resolve) => {
                    setTimeout(resolve, ms);
                });
            }
            var response = await client.send(command);
        }
        console.log('Collection successfully created:');
        console.log(response['collectionDetails']);
```

```
        // Extract the collection endpoint from the response
        var host = (response.collectionDetails[0]['collectionEndpoint'])
        // Pass collection endpoint to index document request
        indexDocument(host)
    } catch (error) {
        console.error(error);
    };
}

async function indexDocument(host) {

    var client = new Client({
        node: host,
        Connection: class extends Connection {
            buildRequestObject(params) {
                var request = super.buildRequestObject(params)
                request.service = 'aoss';
                request.region = 'us-east-1'; // e.g. us-east-1
                var body = request.body;
                request.body = undefined;
                delete request.headers['content-length'];
                request.headers['x-amz-content-sha256'] = 'UNSIGNED-PAYLOAD';
                request = aws4.sign(request, AWS.config.credentials);
                request.body = body;

                return request
            }
        }
    });

    // Create an index
    try {
        var index_name = "sitcoms-eighties";

        var response = await client.indices.create({
            index: index_name
        });

        console.log("Creating index:");
        console.log(response.body);

        // Add a document to the index
        var document = "{ \"title\": \"Seinfeld\", \"creator\": \"Larry David\", \"year
\": \"1989\" }\n";
```

```
        var response = await client.index({
            index: index_name,
            body: document
        });

        console.log("Adding document:");
        console.log(response.body);
    } catch (error) {
        console.error(error);
    };
}

execute()
```

Menggunakan AWS CloudFormation untuk membuat koleksi Amazon OpenSearch Tanpa Server

Anda dapat menggunakannya AWS CloudFormation untuk membuat sumber daya Amazon OpenSearch Tanpa Server seperti pengumpulan, kebijakan keamanan, dan titik akhir VPC. Untuk CloudFormation referensi OpenSearch Tanpa Server yang komprehensif, lihat Amazon OpenSearch Tanpa Server di Panduan Pengguna. AWS CloudFormation

Contoh CloudFormation template berikut membuat kebijakan akses data sederhana, kebijakan jaringan, dan kebijakan keamanan, serta koleksi yang cocok. Ini adalah cara yang baik untuk bangun dan berjalan cepat dengan Amazon OpenSearch Tanpa Server dan menyediakan elemen yang diperlukan untuk membuat dan menggunakan koleksi.

> **Important**
>
> Contoh ini menggunakan akses jaringan publik, yang tidak disarankan untuk beban kerja produksi. Sebaiknya gunakan akses VPC untuk melindungi koleksi Anda. Selengkapnya, lihat AWS::OpenSearchServerless::VpcEndpoint dan the section called “Titik akhir VPC”.

```
AWSTemplateFormatVersion: 2010-09-09
Description: 'Amazon OpenSearch Serverless template to create an IAM user, encryption
 policy, data access policy and collection'
Resources:
  IAMUSer:
```

```
    Type: 'AWS::IAM::User'
    Properties:
      UserName:  aossadmin
  DataAccessPolicy:
    Type: 'AWS::OpenSearchServerless::AccessPolicy'
    Properties:
      Name: quickstart-access-policy
      Type: data
      Description: Access policy for quickstart collection
      Policy: !Sub >-
        [{"Description":"Access for cfn user","Rules":
[{"ResourceType":"index","Resource":["index/*/*"],"Permission":["aoss:*"]},
        {"ResourceType":"collection","Resource":["collection/quickstart"],"Permission":
["aoss:*"]}],
        "Principal":["arn:aws:iam::${AWS::AccountId}:user/aossadmin"]}]
  NetworkPolicy:
    Type: 'AWS::OpenSearchServerless::SecurityPolicy'
    Properties:
      Name: quickstart-network-policy
      Type: network
      Description: Network policy for quickstart collection
      Policy: >-
        [{"Rules":[{"ResourceType":"collection","Resource":["collection/
quickstart"]}, {"ResourceType":"dashboard","Resource":["collection/
quickstart"]}],"AllowFromPublic":true}]
  EncryptionPolicy:
    Type: 'AWS::OpenSearchServerless::SecurityPolicy'
    Properties:
      Name: quickstart-security-policy
      Type: encryption
      Description: Encryption policy for quickstart collection
      Policy: >-
        {"Rules":[{"ResourceType":"collection","Resource":["collection/
quickstart"]}],"AWSOwnedKey":true}
  Collection:
    Type: 'AWS::OpenSearchServerless::Collection'
    Properties:
      Name: quickstart
      Type: TIMESERIES
      Description: Collection to holds timeseries data
    DependsOn: EncryptionPolicy
Outputs:
  IAMUser:
    Value: !Ref IAMUSer
```

```
  DashboardURL:
    Value: !GetAtt Collection.DashboardEndpoint
  CollectionARN:
    Value: !GetAtt Collection.Arn
```

Mengelola batas kapasitas untuk Amazon Tanpa OpenSearch Server

Dengan Amazon OpenSearch Serverless, Anda tidak perlu mengelola kapasitas sendiri. OpenSearch Tanpa server secara otomatis menskalakan kapasitas komputasi untuk akun Anda berdasarkan beban kerja saat ini. Kapasitas komputasi tanpa server diukur dalam OpenSearch Compute Units (OCU). Setiap OCU adalah kombinasi dari 6 GiB memori dan CPU virtual yang sesuai (vCPU), serta transfer data ke Amazon S3. Untuk informasi selengkapnya tentang arsitektur terpisah di Tanpa OpenSearch Server, lihat. the section called “Cara kerjanya”

Saat Anda membuat koleksi pertama Anda, OpenSearch Serverless membuat instance total empat OCU (dua untuk pengindeksan dan dua untuk pencarian). OCU ini selalu ada, bahkan ketika tidak ada aktivitas pengindeksan atau pencarian. Semua koleksi berikutnya dapat membagikan OCU ini (kecuali untuk koleksi dengan AWS KMS kunci unik, yang membuat instance set empat OCU mereka sendiri). Jika diperlukan, OpenSearch Serverless secara otomatis menskalakan dan menambahkan OCU tambahan seiring bertambahnya penggunaan pengindeksan dan penelusuran Anda. Ketika lalu lintas di titik akhir koleksi Anda berkurang, skala kapasitas kembali ke jumlah minimum OCU yang diperlukan untuk ukuran data Anda. Paling-paling, ini akan menurunkan skala menjadi 2 OCU untuk pengindeksan dan 2 OCU untuk pencarian.

Untuk koleksi pencarian dan pencarian vektor, semua data disimpan pada indeks panas untuk memastikan waktu respons kueri yang cepat. Koleksi deret waktu menggunakan kombinasi penyimpanan panas dan hangat, menyimpan data terbaru dalam penyimpanan panas untuk mengoptimalkan waktu respons kueri untuk data yang lebih sering diakses. Untuk informasi selengkapnya, lihat the section called “Memilih jenis koleksi”.

Untuk mengelola kapasitas koleksi Anda dan untuk mengontrol biaya, Anda dapat menentukan pengindeksan maksimum keseluruhan dan kapasitas pencarian untuk akun saat ini dan Wilayah, dan OpenSearch Tanpa Server menskalakan sumber daya koleksi Anda secara otomatis berdasarkan spesifikasi ini.

Karena skala kapasitas pengindeksan dan penelusuran secara terpisah, Anda menentukan batas tingkat akun untuk masing-masing:

- Kapasitas pengindeksan maksimum — OpenSearch Tanpa server dapat meningkatkan kapasitas pengindeksan hingga jumlah OCU ini.
- Kapasitas pencarian maksimum — OpenSearch Tanpa server dapat meningkatkan kapasitas pencarian hingga jumlah OCU ini.

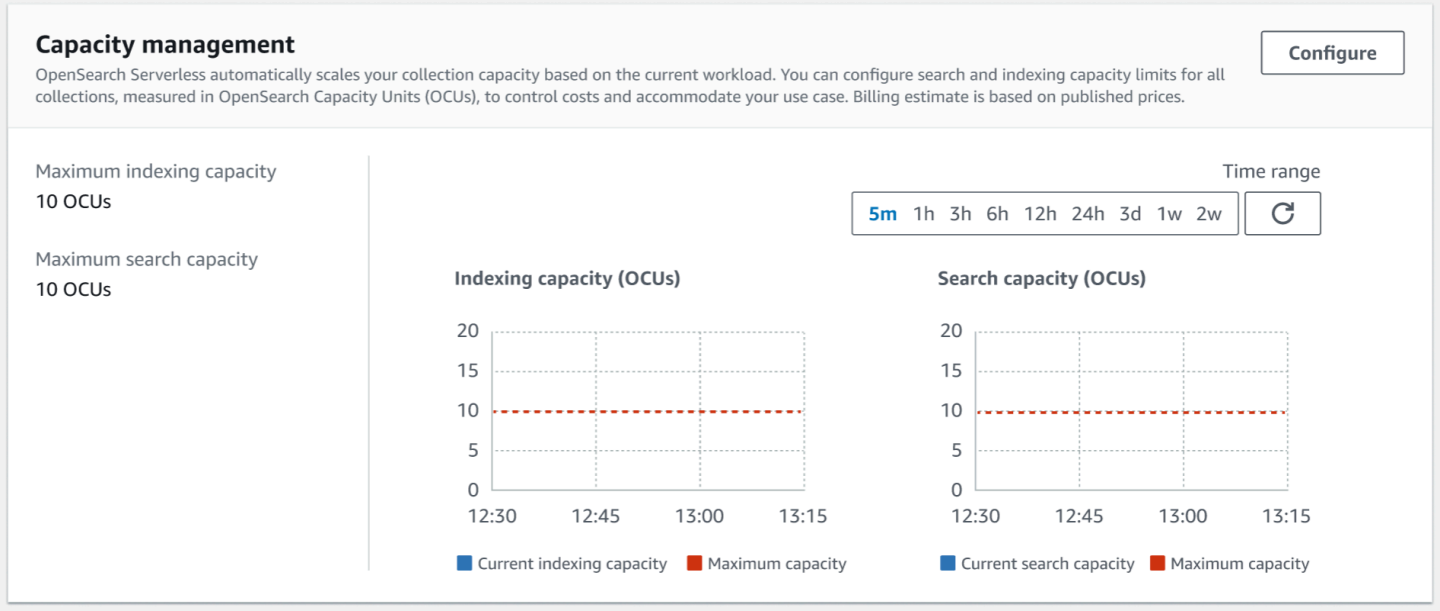
> **Note**
>
> Pada saat ini, pengaturan kapasitas hanya berlaku di tingkat akun. Anda tidak dapat mengonfigurasi batas kapasitas per koleksi.

Tujuan Anda harus memastikan bahwa kapasitas maksimum cukup tinggi untuk menangani lonjakan beban kerja. Berdasarkan pengaturan Anda, OpenSearch Tanpa Server secara otomatis mengukur jumlah OCU untuk koleksi Anda guna memproses beban kerja pengindeksan dan penelusuran.

Topik

Mengkonfigurasi pengaturan kapasitas

Untuk mengonfigurasi pengaturan kapasitas di konsol OpenSearch Tanpa Server, perluas Tanpa Server di panel navigasi kiri dan pilih Dasbor. Tentukan pengindeksan maksimum dan kapasitas pencarian di bawah Manajemen kapasitas:



Untuk mengkonfigurasi kapasitas menggunakanAWS CLI, kirim UpdateAccountSettingspermintaan:

```
aws opensearchserverless update-account-settings \
    --capacity-limits '{ "maxIndexingCapacityInOCU": 8,"maxSearchCapacityInOCU": 9 }'
```

Batas kapasitas maksimum

Untuk ketiga jenis koleksi, kapasitas maksimum default adalah 10 OCU untuk pengindeksan dan 10 OCU untuk pencarian. Kapasitas minimum yang diizinkan untuk akun adalah 2 OCU untuk pengindeksan dan 2 OCU untuk pencarian. Untuk semua koleksi, kapasitas maksimum yang diizinkan adalah 200 OCU untuk pengindeksan dan 200 OCU untuk pencarian. Anda dapat mengonfigurasi jumlah OCU menjadi angka apa pun dari 2 hingga kapasitas maksimum yang diizinkan, dalam kelipatan 2.

Setiap OCU mencakup penyimpanan singkat panas yang cukup untuk 120 GiB data indeks. OpenSearch Tanpa server mendukung hingga 1 TiB data per indeks dalam koleksi pencarian dan pencarian vektor, dan 10 TiB data panas per indeks dalam koleksi deret waktu. Untuk koleksi deret waktu, Anda masih dapat menelan lebih banyak data, yang dapat disimpan sebagai data hangat di S3.

Untuk daftar semua kuota, lihat Kuota tanpa OpenSearch server.

Pemantauan penggunaan kapasitas

Anda dapat memantau CloudWatch metrik `Search0CU` dan `Indexing0CU` tingkat akun untuk memahami bagaimana penskalaan koleksi Anda. Kami menyarankan Anda mengonfigurasi alarm untuk memberi tahu Anda jika akun Anda mendekati ambang batas untuk metrik yang terkait dengan kapasitas, sehingga Anda dapat menyesuaikan pengaturan kapasitas sesuai dengan itu.

Anda juga dapat menggunakan metrik ini untuk menentukan apakah pengaturan kapasitas maksimum Anda sesuai, atau apakah Anda perlu menyesuaikannya. Analisis metrik ini untuk memfokuskan upaya Anda mengoptimalkan efisiensi koleksi Anda. Untuk informasi selengkapnya tentang metrik yang dikirimkan OpenSearch Tanpa Server, lihat. CloudWatch the section called "Pemantauan Tanpa OpenSearch Server"

Menyerap data ke dalam koleksi Amazon Tanpa OpenSearch Server

Bagian ini memberikan detail tentang saluran pipa konsumsi yang didukung untuk konsumsi data ke dalam koleksi Amazon OpenSearch Tanpa Server. Mereka juga mencakup beberapa klien yang dapat Anda gunakan untuk berinteraksi dengan operasi OpenSearch API. Klien Anda harus kompatibel dengan OpenSearch 2.x untuk berintegrasi dengan Tanpa OpenSearch Server.

Topik

- Izin minimum yang diperlukan
- OpenSearch Tertelan
- Fluent Bit
- Amazon Data Firehose
- Lancar
- Go
- Java
- JavaScript
- Logstash
- Python
- Ruby

- Menandatangani permintaan HTTP dengan klien lain

Izin minimum yang diperlukan

Untuk memasukkan data ke dalam koleksi OpenSearch Tanpa Server, prinsipal yang menulis data harus memiliki izin minimum berikut yang ditetapkan dalam kebijakan akses data:

```
[
   {
      "Rules":[
         {
            "ResourceType":"index",
            "Resource":[
               "index/target-collection/logs"
            ],
            "Permission":[
               "aoss:CreateIndex",
               "aoss:WriteDocument",
               "aoss:UpdateIndex"
            ]
         }
      ],
      "Principal":[
         "arn:aws:iam::123456789012:user/my-user"
      ]
   }
]
```

Izin bisa lebih luas jika Anda berencana untuk menulis ke indeks tambahan. *Misalnya, daripada menentukan indeks target tunggal, Anda dapat mengizinkan izin untuk semua indeks (index/ target-collection /*), atau subset indeks (index/ target-collection/logs*).*

Untuk referensi semua operasi OpenSearch API yang tersedia dan izin terkait, lihatthe section called “Operasi dan plugin yang didukung”.

OpenSearch Tertelan

Daripada menggunakan klien pihak ketiga untuk mengirim data langsung ke koleksi OpenSearch Tanpa Server, Anda dapat menggunakan Amazon OpenSearch Ingestion. Anda mengonfigurasi produsen data Anda untuk mengirim data ke OpenSearch Ingestion, dan secara otomatis

mengirimkan data ke koleksi yang Anda tentukan. Anda juga dapat mengonfigurasi OpenSearch Ingestion untuk mengubah data Anda sebelum mengirimkannya. Untuk informasi selengkapnya, lihat OpenSearch Tertelan Amazon.

Pipeline OpenSearch Ingestion memerlukan izin untuk menulis ke koleksi OpenSearch Tanpa Server yang dikonfigurasi sebagai wastafelnya. Izin ini mencakup kemampuan untuk mendeskripsikan koleksi dan mengirim permintaan HTTP ke dalamnya.

Pertama, buat peran IAM yang memiliki `aoss:BatchGetCollection` dan `aoss:APIAccessAll` izin terhadap semua sumber daya ()*. Kemudian, sertakan peran ini dalam kebijakan akses data dan berikan izin untuk membuat indeks, memperbarui indeks, mendeskripsikan indeks, dan menulis dokumen dalam koleksi. Terakhir, tentukan peran ARN sebagai nilai opsi sts_role_arn dalam konfigurasi pipeline.

Untuk instruksi cara menyelesaikan setiap langkah ini, lihat the section called “Memberikan akses jaringan pipa ke koleksi”.

Untuk memulai dengan OpenSearch Ingestion, lihat. the section called “Tutorial: Menyerap data ke dalam koleksi”

Fluent Bit

Anda dapat menggunakan AWS gambar Fluent Bit dan plugin OpenSearch output untuk mencerna data ke dalam koleksi Tanpa OpenSearch Server.

> **Note**
>
> Anda harus memiliki versi 2.30.0 atau yang lebih baru dari image AWS for Fluent Bit agar dapat diintegrasikan dengan Tanpa Server. OpenSearch

Contoh konfigurasi:

Bagian keluaran sampel dari file konfigurasi ini menunjukkan cara menggunakan koleksi OpenSearch Tanpa Server sebagai tujuan. Penambahan penting adalah `AWS_Service_Name` parameter, yaitu`aoss`. `Host`adalah titik akhir koleksi.

```
[OUTPUT]
    Name  opensearch
    Match *
    Host  collection-endpoint.us-west-2.aoss.amazonaws.com
```

```
    Port  443
    Index  my_index
    Trace_Error On
    Trace_Output On
    AWS_Auth On
    AWS_Region <region>
    AWS_Service_Name aoss
    tls     On
    Suppress_Type_Name On
```

Amazon Data Firehose

Firehose mendukung OpenSearch Tanpa Server sebagai tujuan pengiriman. Untuk petunjuk pengiriman data ke OpenSearch Tanpa Server, lihat Membuat Aliran Pengiriman Firehose Data Kinesis dan OpenSearch Pilih Tanpa Server untuk Tujuan Anda di Panduan Pengembang Amazon Data Firehose.

Peran IAM yang Anda berikan kepada Firehose untuk pengiriman harus ditentukan dalam kebijakan akses data dengan `aoss:WriteDocument` izin minimum untuk pengumpulan target, dan Anda harus memiliki indeks yang sudah ada sebelumnya untuk mengirim data. Untuk informasi selengkapnya, lihat the section called “Izin minimum yang diperlukan”.

Sebelum mengirim data ke OpenSearch Tanpa Server, Anda mungkin perlu melakukan transformasi pada data. Untuk mempelajari selengkapnya tentang penggunaan fungsi Lambda untuk menjalankan tugas ini, lihat Transformasi Data Amazon Kinesis Data Firehose dalam panduan yang sama.

Lancar

Anda dapat menggunakan OpenSearch plugin Fluentd untuk mengumpulkan data dari infrastruktur, wadah, dan perangkat jaringan Anda dan mengirimkannya ke koleksi Tanpa OpenSearch Server. Calyptia mempertahankan distribusi Fluentd yang berisi semua dependensi hilir Ruby dan SSL.

Untuk menggunakan Fluentd untuk mengirim data ke Tanpa Server OpenSearch

1. Unduh versi 1.4.2 atau yang lebih baru dari Calyptia Fluentd dari https://www.fluentd.org/download. Versi ini menyertakan OpenSearch plugin secara default, yang mendukung Tanpa OpenSearch Server.
2. Instal paket . Ikuti petunjuk dalam dokumentasi Fluentd berdasarkan sistem operasi Anda:

    - Red Hat Enterprise Linux /CentOs/Amazon Linux

- Debian/Ubuntu
- Windows
- MacOSX

3. Tambahkan konfigurasi yang mengirimkan data ke Tanpa OpenSearch Server. Konfigurasi sampel ini mengirimkan pesan “test” ke satu koleksi. Pastikan untuk melakukan hal berikut:
   - Untuk`host`, tentukan titik akhir koleksi Tanpa OpenSearch Server Anda.
   - Untuk `aws_service_name`, tentukan `aoss`.

```
<source>
@type sample
tag test
test {"hello":"world"}
</source>

<match test>
@type opensearch
host https://collection-endpoint.us-east-1.aoss.amazonaws.com
port 443
index_name fluentd
aws_service_name aoss
</match>
```

4. Jalankan Calyptia Fluentd untuk mulai mengirim data ke koleksi. Misalnya, di Mac Anda dapat menjalankan perintah berikut:

```
sudo launchctl load /Library/LaunchDaemons/calyptia-fluentd.plist
```

Go

Kode contoh berikut menggunakan klien opensearch-go untuk Go untuk membuat koneksi aman ke koleksi OpenSearch Tanpa Server yang ditentukan dan membuat indeks tunggal. Anda harus memberikan nilai untuk `region` dan `host`.

```
package main
```

```
import (
  "context"
  "log"
  "strings"
  "github.com/aws/aws-sdk-go-v2/aws"
  "github.com/aws/aws-sdk-go-v2/config"
  opensearch "github.com/opensearch-project/opensearch-go/v2"
  opensearchapi "github.com/opensearch-project/opensearch-go/v2/opensearchapi"
  requestsigner "github.com/opensearch-project/opensearch-go/v2/signer/awsv2"
)

const endpoint = "" // serverless collection endpoint

func main() {
 ctx := context.Background()

 awsCfg, err := config.LoadDefaultConfig(ctx,
  config.WithRegion("<AWS_REGION>"),
  config.WithCredentialsProvider(
   getCredentialProvider("<AWS_ACCESS_KEY>", "<AWS_SECRET_ACCESS_KEY>",
 "<AWS_SESSION_TOKEN>"),
  ),
 )
 if err != nil {
  log.Fatal(err) // don't log.fatal in a production-ready app
 }

 // create an AWS request Signer and load AWS configuration using default config folder
 or env vars.
 signer, err := requestsigner.NewSignerWithService(awsCfg, "aoss") // "aoss" for Amazon
 OpenSearch Serverless
 if err != nil {
  log.Fatal(err) // don't log.fatal in a production-ready app
 }

 // create an opensearch client and use the request-signer
 client, err := opensearch.NewClient(opensearch.Config{
  Addresses: []string{endpoint},
  Signer:    signer,
 })
 if err != nil {
  log.Fatal("client creation err", err)
 }
```

```
 indexName := "go-test-index"

  // define index mapping
 mapping := strings.NewReader(`{
  "settings": {
    "index": {
         "number_of_shards": 4
         }
       }
  }`)

 // create an index
 createIndex := opensearchapi.IndicesCreateRequest{
  Index: indexName,
    Body: mapping,
 }
 createIndexResponse, err := createIndex.Do(context.Background(), client)
 if err != nil {
  log.Println("Error ", err.Error())
  log.Println("failed to create index ", err)
  log.Fatal("create response body read err", err)
 }
 log.Println(createIndexResponse)

 // delete the index
 deleteIndex := opensearchapi.IndicesDeleteRequest{
  Index: []string{indexName},
 }

 deleteIndexResponse, err := deleteIndex.Do(context.Background(), client)
 if err != nil {
  log.Println("failed to delete index ", err)
  log.Fatal("delete index response body read err", err)
 }
 log.Println("deleting index", deleteIndexResponse)
}

func getCredentialProvider(accessKey, secretAccessKey, token string)
 aws.CredentialsProviderFunc {
 return func(ctx context.Context) (aws.Credentials, error) {
  c := &aws.Credentials{
   AccessKeyID:     accessKey,
   SecretAccessKey: secretAccessKey,
   SessionToken:    token,
```

```
  }
  return *c, nil
 }
}
```

Java

Kode contoh berikut menggunakan klien opensearch-java untuk Java untuk membuat koneksi aman ke koleksi OpenSearch Serverless tertentu dan membuat indeks tunggal. Anda harus memberikan nilai untuk `region` dan `host`.

Perbedaan penting dibandingkan dengan domain OpenSearch Layanan adalah nama layanan (`aoss`bukanes).

```
// import OpenSearchClient to establish connection to OpenSearch Serverless collection
import org.opensearch.client.opensearch.OpenSearchClient;

SdkHttpClient httpClient = ApacheHttpClient.builder().build();

// create an opensearch client and use the request-signer
OpenSearchClient client = new OpenSearchClient(
    new AwsSdk2Transport(
        httpClient,
        "...us-west-2.aoss.amazonaws.com", // serverless collection endpoint
        "aoss" // signing service name
        Region.US_WEST_2, // signing service region
        AwsSdk2TransportOptions.builder().build()
    )
);

String index = "sample-index";

// create an index
CreateIndexRequest createIndexRequest = new
 CreateIndexRequest.Builder().index(index).build();
CreateIndexResponse createIndexResponse = client.indices().create(createIndexRequest);
System.out.println("Create index reponse: " + createIndexResponse);

// delete the index
DeleteIndexRequest deleteIndexRequest = new
 DeleteIndexRequest.Builder().index(index).build();
DeleteIndexResponse deleteIndexResponse = client.indices().delete(deleteIndexRequest);
System.out.println("Delete index reponse: " + deleteIndexResponse);
```

```
httpClient.close();
```

JavaScript

Kode contoh berikut menggunakan klien opensearch-js untuk membuat koneksi aman JavaScript ke koleksi OpenSearch Tanpa Server yang ditentukan, membuat indeks tunggal, menambahkan dokumen, dan menghapus indeks. Anda harus memberikan nilai untuk `node` dan `region`.

Perbedaan penting dibandingkan dengan domain OpenSearch Layanan adalah nama layanan (`aoss`bukanes).

Version 3

Contoh ini menggunakan SDK versi 3 untuk JavaScript di Node.js.

```
const { defaultProvider } = require('@aws-sdk/credential-provider-node');
const { Client } = require('@opensearch-project/opensearch');
const { AwsSigv4Signer } = require('@opensearch-project/opensearch/aws');

async function main() {
    // create an opensearch client and use the request-signer
    const client = new Client({
        ...AwsSigv4Signer({
            region: 'us-west-2',
            service: 'aoss',
            getCredentials: () => {
                const credentialsProvider = defaultProvider();
                return credentialsProvider();
            },
        }),
        node: '' # // serverless collection endpoint
    });

    const index = 'movies';

    // create index if it doesn't already exist
    if (!(await client.indices.exists({ index })).body) {
        console.log((await client.indices.create({ index })).body);
    }

    // add a document to the index
```

```
    const document = { foo: 'bar' };
    const response = await client.index({
        id: '1',
        index: index,
        body: document,
    });
    console.log(response.body);

    // delete the index
    console.log((await client.indices.delete({ index })).body);
}

main();
```

Version 2

Contoh ini menggunakan SDK versi 2 untuk JavaScript di Node.js.

```
const AWS = require('aws-sdk');
const { Client } = require('@opensearch-project/opensearch');
const { AwsSigv4Signer } = require('@opensearch-project/opensearch/aws');

async function main() {
    // create an opensearch client and use the request-signer
    const client = new Client({
        ...AwsSigv4Signer({
            region: 'us-west-2',
            service: 'aoss',
            getCredentials: () =>
                new Promise((resolve, reject) => {
                    AWS.config.getCredentials((err, credentials) => {
                        if (err) {
                            reject(err);
                        } else {
                            resolve(credentials);
                        }
                    });
                }),
        }),
        node: '' # // serverless collection endpoint
    });

    const index = 'movies';
```

```
    // create index if it doesn't already exist
    if (!(await client.indices.exists({ index })).body) {
        console.log((await client.indices.create({
            index
        })).body);
    }

    // add a document to the index
    const document = {
        foo: 'bar'
    };
    const response = await client.index({
        id: '1',
        index: index,
        body: document,
    });
    console.log(response.body);

    // delete the index
    console.log((await client.indices.delete({ index })).body);
}

main();
```

Logstash

Anda dapat menggunakan OpenSearch plugin Logstash untuk mempublikasikan log ke koleksi Tanpa OpenSearch Server.

Untuk menggunakan Logstash untuk mengirim data ke Tanpa Server OpenSearch

1. Instal logstash-output-opensearchplugin versi 2.0.0 atau yang lebih baru menggunakan Docker atau Linux.

   Docker

   Docker meng-host perangkat lunak Logstash OSS dengan plugin output yang sudah diinstal sebelumnya: opensearchproject/ OpenSearch -output-plugin. logstash-oss-with-opensearch
   Anda dapat menarik gambar seperti gambar lainnya:

   ```
   docker pull opensearchproject/logstash-oss-with-opensearch-output-plugin:latest
   ```

Linux

Pertama, instal versi terbaru Logstash jika Anda belum melakukannya. Kemudian, instal versi 2.0.0 dari plugin output:

```
cd logstash-8.5.0/
bin/logstash-plugin install --version 2.0.0 logstash-output-opensearch
```

Jika plugin sudah diinstal, perbarui ke versi terbaru:

```
bin/logstash-plugin update logstash-output-opensearch
```

Dimulai dengan versi 2.0.0 plugin, AWS SDK menggunakan versi 3. Jika Anda menggunakan versi Logstash lebih awal dari 8.4.0, Anda harus menghapus plugin pra-instal dan menginstal AWS plugin: `logstash-integration-aws`

```
/usr/share/logstash/bin/logstash-plugin remove logstash-input-s3
/usr/share/logstash/bin/logstash-plugin remove logstash-input-sqs
/usr/share/logstash/bin/logstash-plugin remove logstash-output-s3
/usr/share/logstash/bin/logstash-plugin remove logstash-output-sns
/usr/share/logstash/bin/logstash-plugin remove logstash-output-sqs
/usr/share/logstash/bin/logstash-plugin remove logstash-output-cloudwatch

/usr/share/logstash/bin/logstash-plugin install --version 0.1.0.pre logstash-
integration-aws
```

2. Agar plugin OpenSearch output berfungsi dengan OpenSearch Tanpa Server, Anda harus membuat modifikasi berikut pada bagian `opensearch` output logstash.conf:

   - Tentukan `aoss` sebagai di `service_name` bawah`auth_type`.
   - Tentukan titik akhir koleksi Anda untuk`hosts`.
   - Tambahkan parameter `default_server_major_version` dan`legacy_template`. Parameter ini diperlukan agar plugin dapat bekerja dengan Tanpa OpenSearch Server.

```
output {
  opensearch {
    hosts => "collection-endpoint:443"
    auth_type => {
```

```
      ...
      service_name => 'aoss'
    }
    default_server_major_version => 2
    legacy_template => false
  }
}
```

Contoh file konfigurasi ini mengambil inputnya dari file dalam bucket S3 dan mengirimkannya ke koleksi Tanpa OpenSearch Server:

```
input {
  s3  {
    bucket => "my-s3-bucket"
    region => "us-east-1"
  }
}

output {
  opensearch {
    ecs_compatibility => disabled
    hosts => "https://my-collection-endpoint.us-east-1.aoss.amazonaws.com:443"
    index => my-index
    auth_type => {
      type => 'aws_iam'
      aws_access_key_id => 'your-access-key'
      aws_secret_access_key => 'your-secret-key'
      region => 'us-east-1'
      service_name => 'aoss'
    }
    default_server_major_version => 2
    legacy_template => false
  }
}
```

3. Kemudian, jalankan Logstash dengan konfigurasi baru untuk menguji plugin:

```
bin/logstash -f config/test-plugin.conf
```

Python

Kode contoh berikut menggunakan klien opensearch-py untuk Python untuk membuat koneksi aman ke koleksi OpenSearch Tanpa Server yang ditentukan, membuat indeks tunggal, dan mencari indeks itu. Anda harus memberikan nilai untuk `region` dan `host`.

Perbedaan penting dibandingkan dengan domain OpenSearch Layanan adalah nama layanan (`aossbukanes`).

```
from opensearchpy import OpenSearch, RequestsHttpConnection, AWSV4SignerAuth
import boto3

host = ''  # serverless collection endpoint, without https://
region = ''  # e.g. us-east-1

service = 'aoss'
credentials = boto3.Session().get_credentials()
auth = AWSV4SignerAuth(credentials, region, service)

# create an opensearch client and use the request-signer
client = OpenSearch(
    hosts=[{'host': host, 'port': 443}],
    http_auth=auth,
    use_ssl=True,
    verify_certs=True,
    connection_class=RequestsHttpConnection,
    pool_maxsize=20,
)

# create an index
index_name = "books-index"
create_response = client.indices.create(
    index_name
)

print('\nCreating index:')
print(create_response)

# index a document
document = {
  'title': 'The Green Mile,
  'director': 'Stephen King',
  'year': '1996'
```

```
}

response = client.index(
    index = 'books-index',
    body = document,
    id = '1'
)


# delete the index
delete_response = client.indices.delete(
    index_name
)

print('\nDeleting index:')
print(delete_response)
```

Ruby

`opensearch-aws-sigv4`Permata menyediakan akses ke OpenSearch Tanpa Server, bersama dengan OpenSearch Layanan, di luar kotak. Ini memiliki semua fitur klien opensearch-ruby karena merupakan ketergantungan permata ini.

Saat membuat instance Sigv4 signer, tentukan `aoss` sebagai nama layanan:

```
require 'opensearch-aws-sigv4'
require 'aws-sigv4'

signer = Aws::Sigv4::Signer.new(service: 'aoss',
                                region: 'us-west-2',
                                access_key_id: 'key_id',
                                secret_access_key: 'secret')

# create an opensearch client and use the request-signer
client = OpenSearch::Aws::Sigv4Client.new(
  { host: 'https://your.amz-opensearch-serverless.endpoint',
    log: true },
  signer)

# create an index
index = 'prime'
client.indices.create(index: index)
```

```
# insert data
client.index(index: index, id: '1', body: { name: 'Amazon Echo',
                                            msrp: '5999',
                                            year: 2011 })

# query the index
client.search(body: { query: { match: { name: 'Echo' } } })

# delete index entry
client.delete(index: index, id: '1')

# delete the index
client.indices.delete(index: index)
```

Menandatangani permintaan HTTP dengan klien lain

Persyaratan berikut berlaku saat menandatangani permintaan ke koleksi OpenSearch Tanpa Server saat Anda membuat permintaan HTTP dengan klien lain.

- Anda harus menentukan nama layanan sebagai`aoss`.
- `x-amz-content-sha256`Header diperlukan untuk semua permintaan AWS Signature Version 4. Ini menyediakan hash dari payload permintaan. Jika ada payload permintaan, tetapkan nilainya ke hash kriptografi Secure Hash Algorithm (SHA) (SHA256). Jika tidak ada payload permintaan, tetapkan nilainya`e3b0c44298fc1c149afbf4c8996fb92427ae41e4649b934ca495991b7852b855`, yang merupakan hash dari string kosong.

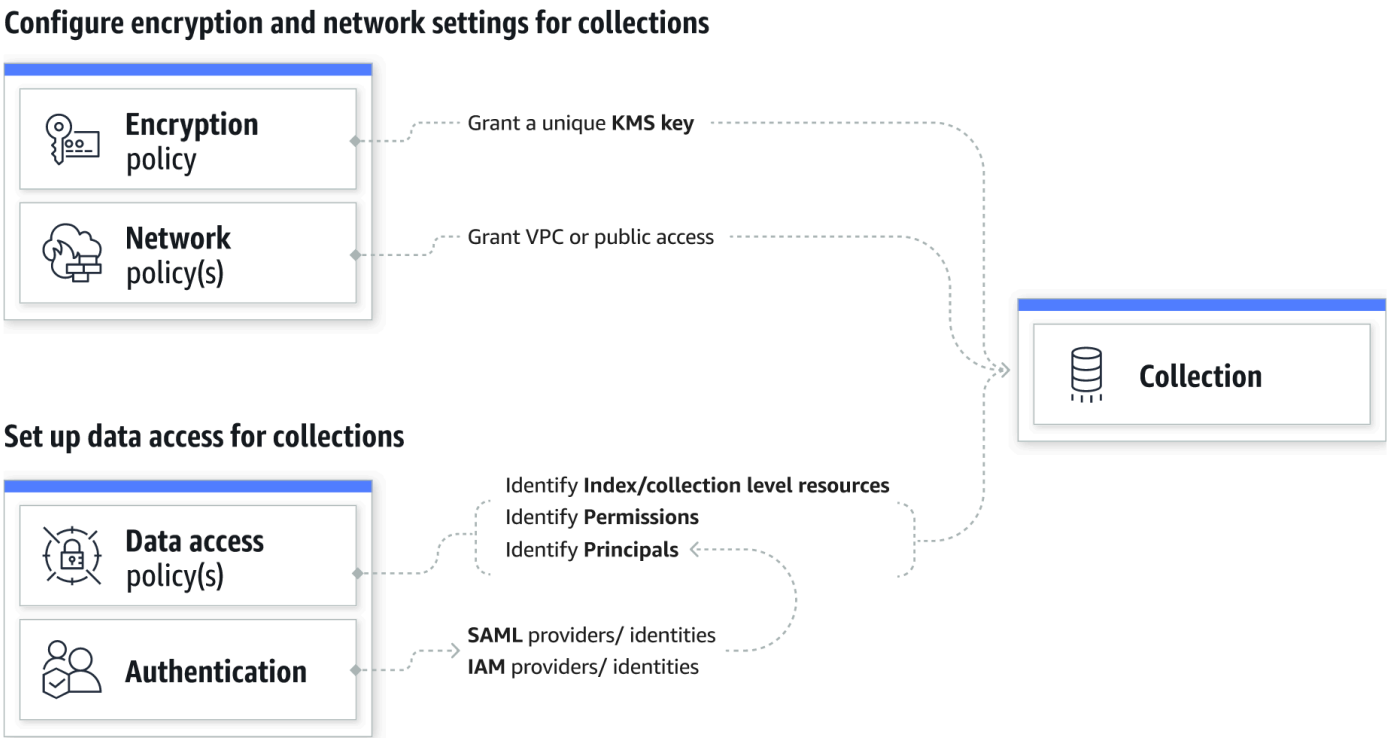
Ikhtisar keamanan di Amazon Tanpa OpenSearch Server

Keamanan di Amazon OpenSearch Tanpa Server berbeda secara mendasar dari keamanan di OpenSearch Layanan Amazon dengan cara berikut:

Fitur	OpenSearch Layanan	OpenSearch Tanpa server
Kontrol akses data	Akses data ditentukan oleh kebijakan IAM dan kontrol akses berbutir halus.	Akses data ditentukan oleh kebijakan akses data.
Enkripsi saat istirahat	Enkripsi saat istirahat adalah opsional untuk domain.	Enkripsi saat istirahat diperlukan untuk koleksi.

Fitur	OpenSearch Layanan	OpenSearch Tanpa server
Pengaturan dan administrasi keamanan	Anda harus mengonfigurasi jaringan, enkripsi, dan akses data secara individual untuk setiap domain.	Anda dapat menggunakan kebijakan keamanan untuk mengelola pengaturan keamanan untuk beberapa koleksi dalam skala besar.

Diagram berikut menggambarkan komponen keamanan yang membentuk koleksi fungsional. Koleksi harus memiliki kunci enkripsi yang ditetapkan, pengaturan akses jaringan, dan kebijakan akses data yang cocok yang memberikan izin ke sumber dayanya.



Topik

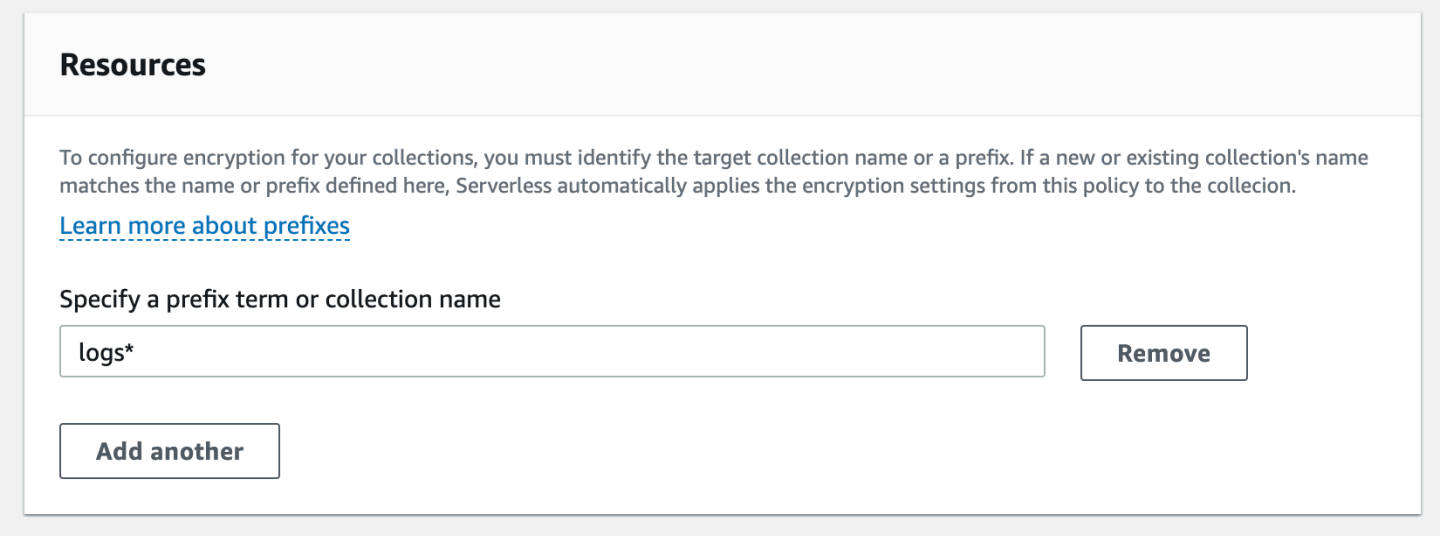
- Kebijakan enkripsi
- Kebijakan jaringan
- Kebijakan akses data
- Autentikasi IAM dan SAMP
- Keamanan infrastruktur

- Memulai dengan keamanan di Amazon Tanpa OpenSearch Server
- Identity and Access Management untuk Amazon OpenSearch Serverless
- Enkripsi di Amazon OpenSearch Tanpa Server
- Akses jaringan untuk Amazon Tanpa OpenSearch Server
- Kontrol akses data untuk Amazon Tanpa OpenSearch Server
- Akses Amazon OpenSearch Tanpa Server menggunakan titik akhir antarmuka () AWS PrivateLink
- Otentikasi SAMP untuk Amazon Tanpa Server OpenSearch
- Validasi kepatuhan untuk Amazon Tanpa Server OpenSearch

Kebijakan enkripsi

Kebijakan enkripsi menentukan apakah koleksi Anda dienkripsi dengan kunci yang dikelola pelanggan Kunci milik AWS atau pelanggan. Kebijakan enkripsi terdiri dari dua komponen: pola sumber daya dan kunci enkripsi. Pola sumber daya menentukan koleksi atau koleksi mana yang berlaku untuk kebijakan tersebut. Kunci enkripsi menentukan bagaimana koleksi terkait akan diamankan.

Untuk menerapkan kebijakan ke beberapa koleksi, Anda menyertakan wildcard (*) dalam aturan kebijakan. Misalnya, kebijakan berikut berlaku untuk semua koleksi dengan nama yang dimulai dengan “log”.

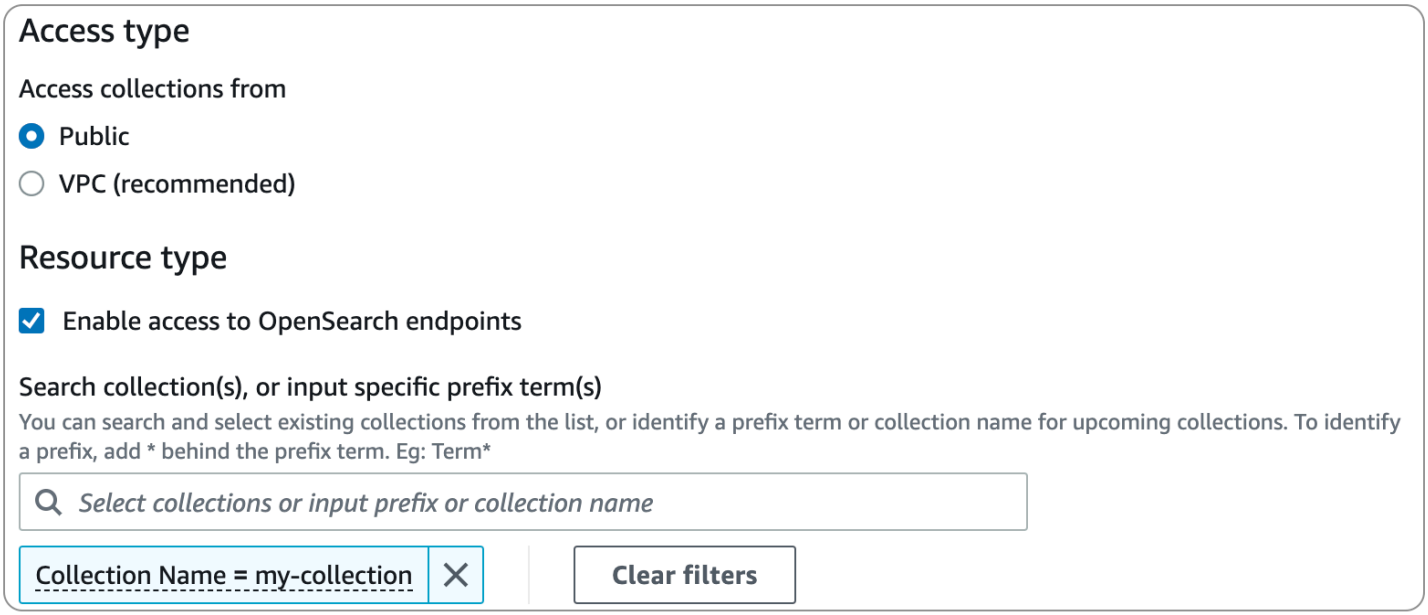


Kebijakan enkripsi merampingkan proses pembuatan dan pengelolaan koleksi, terutama ketika Anda melakukannya secara terprogram. Anda dapat membuat koleksi hanya dengan menentukan nama, dan kunci enkripsi secara otomatis ditetapkan pada saat pembuatan.

Kebijakan jaringan

Kebijakan jaringan menentukan apakah koleksi Anda dapat diakses secara pribadi, atau melalui internet dari jaringan publik. Koleksi pribadi dapat diakses melalui titik akhir VPC yang OpenSearch dikelola tanpa server, atau secara spesifik Layanan AWS seperti Amazon Bedrock menggunakan akses pribadi.Layanan AWS Sama seperti kebijakan enkripsi, kebijakan jaringan dapat diterapkan ke beberapa koleksi, yang memungkinkan Anda mengelola akses jaringan untuk banyak koleksi dalam skala besar.

Kebijakan jaringan terdiri dari dua komponen: tipe akses dan tipe sumber daya. Jenis akses dapat bersifat publik atau pribadi. Jenis sumber daya menentukan apakah akses yang Anda pilih berlaku untuk titik akhir koleksi, titik akhir OpenSearch Dasbor, atau keduanya.



Jika Anda berencana untuk mengonfigurasi akses VPC dalam kebijakan jaringan, Anda harus terlebih dahulu membuat satu atau beberapa titik akhir VPC yang dikelola Tanpa OpenSearch Server. Titik akhir ini memungkinkan Anda mengakses OpenSearch Tanpa Server seolah-olah berada di VPC Anda, tanpa menggunakan gateway internet, perangkat NAT, koneksi VPN, atau koneksi. AWS Direct Connect

Akses pribadi ke hanya Layanan AWS dapat diterapkan ke OpenSearch titik akhir koleksi, bukan ke titik akhir OpenSearch Dasbor. Layanan AWS tidak dapat diberikan akses ke OpenSearch Dasbor.

Kebijakan akses data

Kebijakan akses data menentukan cara pengguna mengakses data dalam koleksi Anda. Kebijakan akses data membantu Anda mengelola koleksi dalam skala besar dengan secara otomatis menetapkan izin akses ke koleksi dan indeks yang cocok dengan pola tertentu. Beberapa kebijakan dapat diterapkan ke satu sumber daya.

Kebijakan akses data terdiri dari seperangkat aturan, masing-masing dengan tiga komponen: jenis sumber daya, sumber daya yang diberikan, dan sekumpulan izin. Jenis sumber daya dapat berupa koleksi atau indeks. Sumber daya yang diberikan dapat berupa nama koleksi/indeks atau pola dengan wildcard (*). Daftar izin menentukan operasi OpenSearch API mana yang dapat diakses oleh kebijakan. Selain itu, kebijakan tersebut berisi daftar kepala sekolah, yang menentukan peran IAM, pengguna, dan identitas SAMP untuk diberikan akses.


Selected principals

Principals

arn:aws:iam::478253424788:user/Administrator

saml/478253424788/myprovider/user/Annie

Granted resources and permissions (2)

Granted resources	Resource type	Permissions
collection/autopartsinventory	collection	aoss:CreateCollectionItems aoss:UpdateCollectionItems
index/test-collection/*	index	aoss:ReadDocument aoss:DescribeIndex

Untuk informasi selengkapnya tentang format kebijakan akses data, lihat sintaks kebijakan.

Sebelum Anda membuat kebijakan akses data, Anda harus memiliki satu atau beberapa peran IAM atau pengguna, atau identitas SAMP, untuk menyediakan akses ke dalam kebijakan. Untuk detailnya, lihat bagian selanjutnya.

Autentikasi IAM dan SAMP

Prinsipal IAM dan identitas SAMP adalah salah satu blok bangunan kebijakan akses data. Dalam `principal` pernyataan kebijakan akses, Anda dapat menyertakan peran IAM, pengguna, dan identitas SAMP. Prinsipal ini kemudian diberikan izin yang Anda tentukan dalam aturan kebijakan terkait.

```
[
   {
```

```
      "Rules":[
         {
            "ResourceType":"index",
            "Resource":[
               "index/marketing/orders*"
            ],
            "Permission":[
               "aoss:*"
            ]
         }
      ],
      "Principal":[
         "arn:aws:iam::123456789012:user/Dale",
         "arn:aws:iam::123456789012:role/RegulatoryCompliance",
         "saml/123456789012/myprovider/user/Annie"
      ]
   }
]
```

Anda mengonfigurasi otentikasi SAMP secara langsung di dalam OpenSearch Tanpa Server. Untuk informasi selengkapnya, lihat the section called “Otentikasi SAMP”.

Keamanan infrastruktur

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

Anda menggunakan panggilan API yang AWS dipublikasikan untuk mengakses Amazon OpenSearch Tanpa Server melalui jaringan. Klien harus mendukung Keamanan Lapisan Pengangkutan (TLS). Kami mensyaratkan TLS 1.2 dan menganjurkan TLS 1.3. Untuk daftar cipher yang didukung untuk TLS 1.3, lihat protokol TLS dan cipher dalam dokumentasi Elastic Load Balancing.

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

Memulai dengan keamanan di Amazon Tanpa OpenSearch Server

Tutorial berikut membantu Anda mulai menggunakan Amazon Tanpa OpenSearch Server. Kedua tutorial menyelesaikan langkah dasar yang sama, tetapi yang satu menggunakan konsol sementara yang lain menggunakan. AWS CLI

Perhatikan bahwa kasus penggunaan dalam tutorial ini disederhanakan. Kebijakan jaringan dan keamanan cukup terbuka. Dalam beban kerja produksi, kami menyarankan Anda mengonfigurasi fitur keamanan yang lebih kuat seperti otentikasi SAMP, akses VPC, dan kebijakan akses data yang membatasi.

Topik

Tutorial: Memulai keamanan di Amazon OpenSearch Tanpa Server (konsol)

Tutorial ini memandu Anda melalui langkah-langkah dasar untuk membuat dan mengelola kebijakan keamanan menggunakan konsol Amazon OpenSearch Tanpa Server.

Anda akan menyelesaikan langkah-langkah berikut dalam tutorial ini:

1. Konfigurasikan izin
2. Buat kebijakan enkripsi
3. Buat kebijakan jaringan
4. Konfigurasikan kebijakan akses data
5. Buat koleksi
6. Unggah dan cari data

Tutorial ini memandu Anda melalui pengaturan koleksi menggunakanAWS Management Console. Untuk langkah yang sama menggunakanAWS CLI, lihatthe section called “Tutorial: Memulai dengan keamanan (CLI)”.

Langkah 1: Konfigurasikan izin

> **Note**
>
> Anda dapat melewati langkah ini jika Anda sudah menggunakan kebijakan berbasis identitas yang lebih luas, seperti atau. `Action":"aoss:*"` `Action":"*"` Namun, di lingkungan produksi, kami menyarankan Anda mengikuti prinsip hak istimewa paling sedikit dan hanya menetapkan izin minimum yang diperlukan untuk menyelesaikan tugas.

Untuk menyelesaikan tutorial ini, Anda harus memiliki izin IAM yang benar. Pengguna atau peran Anda harus memiliki kebijakan berbasis identitas terlampir dengan izin minimum berikut:

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Action": [
        "aoss:ListCollections",
        "aoss:BatchGetCollection",
        "aoss:CreateCollection",
        "aoss:CreateSecurityPolicy",
        "aoss:GetSecurityPolicy",
        "aoss:ListSecurityPolicies",
        "aoss:CreateAccessPolicy",
        "aoss:GetAccessPolicy",
        "aoss:ListAccessPolicies"
      ],
      "Effect": "Allow",
      "Resource": "*"
    }
  ]
}
```

Untuk daftar lengkap izin OpenSearch Tanpa Server, lihat. the section called “Pengelolaan Identitas dan Akses”

Langkah 2: Buat kebijakan enkripsi

Kebijakan enkripsi menentukan AWS KMS kunci yang akan digunakan OpenSearch Tanpa Server untuk mengenkripsi koleksi. Anda dapat mengenkripsi koleksi dengan Kunci yang dikelola AWS atau

kunci yang berbeda. Untuk kesederhanaan dalam tutorial ini, kita akan mengenkripsi koleksi kita dengan fileKunci yang dikelola AWS.

Untuk membuat kebijakan enkripsi

1. Buka konsol OpenSearch Layanan Amazon di https://console.aws.amazon.com/aos/home.
2. Perluas Tanpa Server di panel navigasi kiri dan pilih Kebijakan enkripsi.
3. Pilih Buat kebijakan enkripsi.
4. Beri nama kebijakan buku-buku kebijakan. Untuk deskripsi, masukkan Kebijakan enkripsi untuk koleksi buku.
5. Di bawah Sumber Daya, masukkan buku, yang akan Anda beri nama koleksi Anda. Jika Anda ingin lebih luas, Anda dapat menyertakan tanda bintang (`books*`) untuk membuat kebijakan berlaku untuk semua koleksi yang dimulai dengan kata “buku”.
6. Untuk Enkripsi, tetap pilih Gunakan kunci yang AWS dimiliki.
7. Pilih Buat.

Langkah 3: Buat kebijakan jaringan

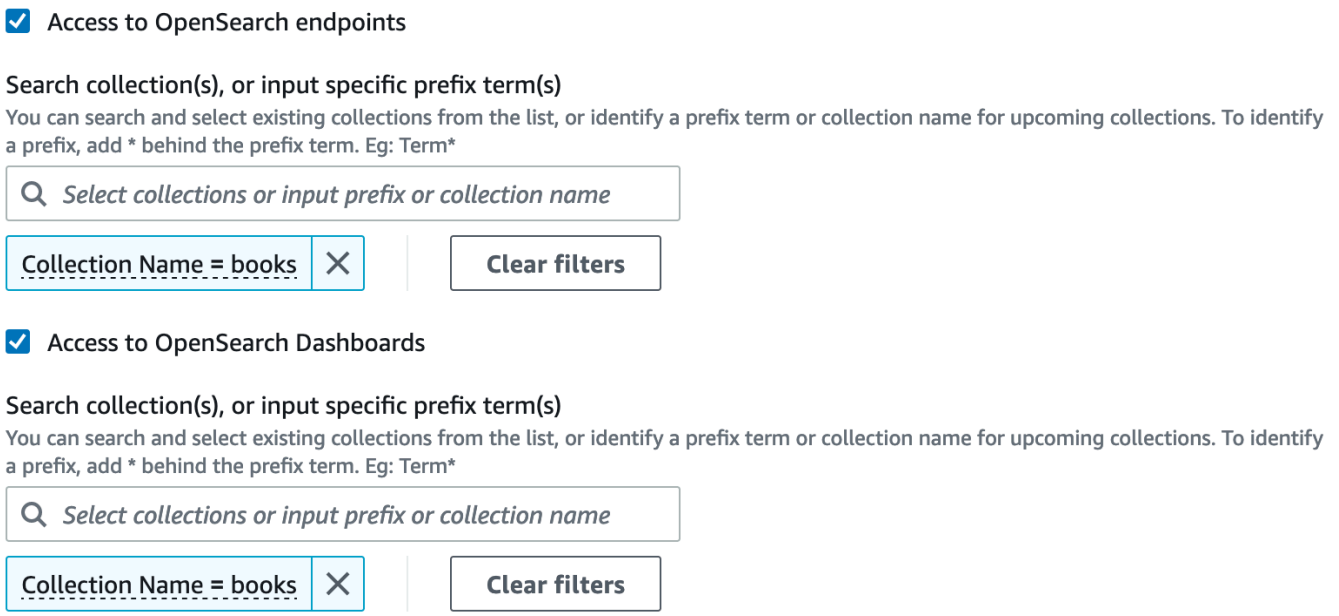
Kebijakan jaringan menentukan apakah koleksi Anda dapat diakses melalui internet dari jaringan publik, atau apakah harus diakses melalui titik akhir VPC yang OpenSearch dikelola Tanpa Server. Dalam tutorial ini, kita akan mengkonfigurasi akses publik.

Untuk membuat kebijakan jaringan

1. Pilih Kebijakan jaringan di panel navigasi kiri dan pilih Buat kebijakan jaringan.
2. Beri nama kebijakan buku-buku kebijakan. Untuk deskripsi, masukkan Kebijakan jaringan untuk koleksi buku.
3. Di bawah Aturan 1, beri nama aturan Akses publik untuk koleksi buku.
4. Untuk kesederhanaan dalam tutorial ini, kita akan mengkonfigurasi akses publik untuk koleksi buku. Untuk jenis akses, pilih Publik.
5. Kita akan mengakses koleksi dari OpenSearch Dashboards. Untuk melakukan ini, Anda perlu mengonfigurasi akses jaringan untuk Dasbor dan OpenSearch titik akhir, jika tidak Dasbor tidak akan berfungsi.

   Untuk jenis sumber daya, aktifkan Access to OpenSearch endpoint dan Access to OpenSearch Dashboards.

6. Di kedua kotak input, masukkan Nama Koleksi = buku. Pengaturan ini mencakup kebijakan sehingga hanya berlaku untuk satu koleksi (`books`). Aturan Anda akan terlihat seperti ini:



7. Pilih Buat.

Langkah 4: Buat kebijakan akses data

Data pengumpulan Anda tidak akan dapat diakses sampai Anda mengonfigurasi akses data. Kebijakan akses data terpisah dari kebijakan berbasis identitas IAM yang Anda konfigurasikan pada langkah 1. Mereka memungkinkan pengguna untuk mengakses data aktual dalam koleksi.

Dalam tutorial ini, kami akan memberikan satu pengguna izin yang diperlukan untuk mengindeks data ke dalam koleksi buku.

Untuk membuat kebijakan akses data

1. Pilih Kebijakan akses data di panel navigasi kiri dan pilih Buat kebijakan akses.
2. Beri nama kebijakan buku-buku kebijakan. Untuk deskripsi, masukkan Kebijakan akses data untuk koleksi buku.
3. Pilih JSON untuk metode definisi kebijakan dan tempel kebijakan berikut di editor JSON.

   Ganti ARN utama dengan ARN akun yang akan Anda gunakan untuk masuk ke OpenSearch Dasbor dan mengindeks data.

```
[
```

```
   {
      "Rules":[
         {
            "ResourceType":"index",
            "Resource":[
               "index/books/*"
            ],
            "Permission":[
               "aoss:CreateIndex",
               "aoss:DescribeIndex",
               "aoss:ReadDocument",
               "aoss:WriteDocument",
               "aoss:UpdateIndex",
               "aoss:DeleteIndex"
            ]
         }
      ],
      "Principal":[
         "arn:aws:iam::123456789012:user/my-user"
      ]
   }
]
```

Kebijakan ini memberi satu pengguna izin minimum yang diperlukan untuk membuat indeks dalam koleksi buku, mengindeks beberapa data, dan mencarinya.

4. Pilih Buat.

Langkah 5: Buat koleksi

Sekarang setelah Anda mengonfigurasi enkripsi dan kebijakan jaringan, Anda dapat membuat koleksi yang cocok dan pengaturan keamanan akan diterapkan secara otomatis padanya.

Untuk membuat koleksi OpenSearch Tanpa Server

1. Pilih Koleksi di panel navigasi kiri dan pilih Buat koleksi.
2. Beri nama buku koleksi.
3. Untuk jenis koleksi, pilih Cari.
4. Di bawah Enkripsi, OpenSearch Tanpa Server memberi tahu Anda bahwa nama koleksi cocok dengan kebijakan enkripsi. `books-policy`

5. Di bawah Pengaturan akses jaringan, OpenSearch Tanpa Server memberi tahu Anda bahwa nama koleksi cocok dengan kebijakan jaringan. `books-policy`
6. Pilih Berikutnya.
7. Di bawah opsi kebijakan akses data, OpenSearch Tanpa Server memberi tahu Anda bahwa nama koleksi cocok dengan kebijakan akses `books-policy` data.
8. Pilih Berikutnya.
9. Tinjau konfigurasi koleksi dan pilih Kirim. Koleksi biasanya membutuhkan waktu kurang dari satu menit untuk diinisialisasi.

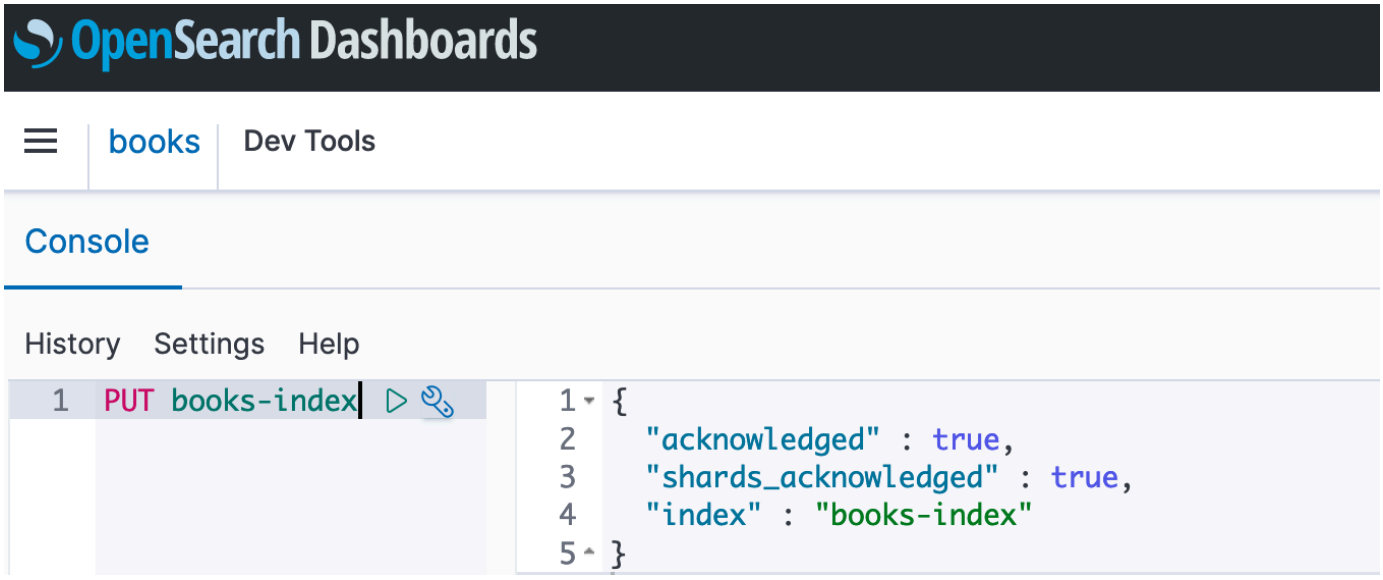
Langkah 6: Unggah dan cari data

Anda dapat mengunggah data ke koleksi OpenSearch Tanpa Server menggunakan Postman atau curl. Untuk singkatnya, contoh-contoh ini menggunakan Dev Tools dalam konsol OpenSearch Dashboards.

Untuk mengindeks dan mencari data dalam koleksi

1. Pilih Koleksi di panel navigasi kiri dan pilih koleksi buku untuk membuka halaman detailnya.
2. Pilih URL OpenSearch Dasbor untuk koleksi. URL mengambil format`https://collection-id.us-east-1.aoss.amazonaws.com/_dashboards`.
3. Masuk ke OpenSearch Dasbor menggunakan AWSakses dan kunci rahasia untuk prinsipal yang Anda tentukan dalam kebijakan akses data Anda.
4. Di dalam OpenSearch Dashboards, buka menu navigasi kiri dan pilih Dev Tools.
5. Untuk membuat indeks tunggal yang disebut books-index, jalankan perintah berikut:

```
PUT books-index
```



6. Untuk mengindeks satu dokumen ke books-index, jalankan perintah berikut:

```
PUT books-index/_doc/1
{
  "title": "The Shining",
  "author": "Stephen King",
  "year": 1977
}
```

7. Untuk mencari data di OpenSearch Dasbor, Anda perlu mengonfigurasi setidaknya satu pola indeks. OpenSearch menggunakan pola-pola ini untuk mengidentifikasi indeks mana yang ingin Anda analisis. Buka menu utama Dashboards, pilih Stack Management, pilih Index Patterns, dan kemudian pilih Create index pattern. Untuk tutorial ini, masukkan books-index.
8. Pilih Langkah selanjutnya dan kemudian pilih Buat pola indeks. Setelah pola dibuat, Anda dapat melihat berbagai bidang dokumen seperti `author` dan `title`.
9. Untuk mulai mencari data Anda, buka menu utama lagi dan pilih Discover, atau gunakan API pencarian.

Tutorial: Memulai keamanan di Amazon OpenSearch Tanpa Server (CLI)

Tutorial ini memandu Anda melalui langkah-langkah yang dijelaskan dalam tutorial memulai konsol untuk keamanan, tetapi menggunakan AWS CLI bukan konsol OpenSearch Layanan.

Anda akan menyelesaikan langkah-langkah berikut dalam tutorial ini:

1. Membuat kebijakan izin IAM
2. Menyelesaikan kebijakan IAM ke peran IAM
3. Buat kebijakan enkripsi
4. Buat kebijakan jaringan
5. Buat koleksi
6. Konfigurasikan kebijakan akses data
7. Ambil titik akhir koleksi
8. Unggah data ke koneksi Anda
9. Cari data dalam koleksi Anda

Tujuan dari tutorial ini adalah untuk mengatur koleksi OpenSearch Serverless tunggal dengan enkripsi, jaringan, dan pengaturan akses data yang cukup sederhana. Misalnya, kami akan mengonfigurasi akses jaringan publik, enkripsi Kunci yang dikelola AWS untuk, dan kebijakan akses data yang disederhanakan yang memberikan izin minimal kepada satu pengguna.

Dalam skenario produksi, pertimbangkan untuk menerapkan konfigurasi yang lebih kuat, termasuk otentikasi SAFL, kunci enkripsi khusus, dan akses VPC.

Untuk memulai dengan kebijakan keamanan di Tanpa OpenSearch Server

1.
> **Note**
>
> Anda dapat melewati langkah ini jika Anda sudah menggunakan kebijakan berbasis identitas yang lebih luas, seperti atau. `Action":"aoss:*"` `Action":"*"` Namun, di lingkungan produksi, kami menyarankan Anda mengikuti prinsip hak istimewa paling sedikit dan hanya menetapkan izin minimum yang diperlukan untuk menyelesaikan tugas.

   Untuk memulai, buat AWS Identity and Access Management kebijakan dengan izin minimum yang diperlukan untuk melakukan langkah-langkah dalam tutorial ini. Kami akan memberi nama kebijakan`TutorialPolicy`:

```
aws iam create-policy \
  --policy-name TutorialPolicy \
```

```
  --policy-document "{\"Version\": \"2012-10-17\",\"Statement\":
 [{\"Action\": [\"aoss:ListCollections\",\"aoss:BatchGetCollection\",
\"aoss:CreateCollection\",\"aoss:CreateSecurityPolicy\",\"aoss:GetSecurityPolicy\",
\"aoss:ListSecurityPolicies\",\"aoss:CreateAccessPolicy\",\"aoss:GetAccessPolicy\",
\"aoss:ListAccessPolicies\"],\"Effect\": \"Allow\",\"Resource\": \"*\"}]}"
```

Sampel respon

```
{
    "Policy": {
        "PolicyName": "TutorialPolicy",
        "PolicyId": "ANPAW6WRAECKG6QJWUV7U",
        "Arn": "arn:aws:iam::123456789012:policy/TutorialPolicy",
        "Path": "/",
        "DefaultVersionId": "v1",
        "AttachmentCount": 0,
        "PermissionsBoundaryUsageCount": 0,
        "IsAttachable": true,
        "CreateDate": "2022-10-16T20:57:18+00:00",
        "UpdateDate": "2022-10-16T20:57:18+00:00"
    }
}
```

2. Lampirkan `TutorialPolicy` ke peran IAM yang akan mengindeks dan mencari data dalam koleksi. Kami akan memberi nama penggunaTutorialRole:

```
aws iam attach-role-policy \
  --role-name TutorialRole \
  --policy-arn arn:aws:iam::123456789012:policy/TutorialPolicy
```

3. Sebelum membuat koleksi, Anda perlu membuat kebijakan enkripsi yang Kunci milik AWS menetapkan koleksi buku yang akan Anda buat di langkah selanjutnya.

   Kirim permintaan berikut untuk membuat kebijakan enkripsi untuk koleksi buku:

```
aws opensearchserverless create-security-policy \
  --name books-policy \
  --type encryption --policy "{\"Rules\":[{\"ResourceType\":\"collection\",
\"Resource\":[\"collection\/books\"]}],\"AWSOwnedKey\":true}"
```

   Sampel respon

```
{
    "securityPolicyDetail": {
        "type": "encryption",
        "name": "books-policy",
        "policyVersion": "MTY2OTI0MDAwNTk5MF8x",
        "policy": {
            "Rules": [
                {
                    "Resource": [
                        "collection/books"
                    ],
                    "ResourceType": "collection"
                }
            ],
            "AWSOwnedKey": true
        },
        "createdDate": 1669240005990,
        "lastModifiedDate": 1669240005990
    }
}
```

4. Buat kebijakan jaringan yang menyediakan akses publik ke koleksi buku:

```
aws opensearchserverless create-security-policy --name books-policy --type network
 \
  --policy "[{\"Description\":\"Public access for books collection\",\"Rules
\":[{\"ResourceType\":\"dashboard\",\"Resource\":[\"collection\/books\"]},
{\"ResourceType\":\"collection\",\"Resource\":[\"collection\/books\"]}],
\"AllowFromPublic\":true}]"
```

Sampel respon

```
{
    "securityPolicyDetail": {
        "type": "network",
        "name": "books-policy",
        "policyVersion": "MTY2OTI0MDI1Njk1NV8x",
        "policy": [
            {
                "Rules": [
                    {
                        "Resource": [
```

```
                        "collection/books"
                    ],
                    "ResourceType": "dashboard"
                },
                {
                    "Resource": [
                        "collection/books"
                    ],
                    "ResourceType": "collection"
                }
            ],
            "AllowFromPublic": true,
            "Description": "Public access for books collection"
        }
    ],
    "createdDate": 1669240256955,
    "lastModifiedDate": 1669240256955
  }
}
```

5. Buat koleksi buku:

```
aws opensearchserverless create-collection --name books --type SEARCH
```

Sampel respon

```
{
    "createCollectionDetail": {
        "id": "8kw362bpwg4gx9b2f6e0",
        "name": "books",
        "status": "CREATING",
        "type": "SEARCH",
        "arn": "arn:aws:aoss:us-
east-1:123456789012:collection/8kw362bpwg4gx9b2f6e0",
        "kmsKeyArn": "auto",
        "createdDate": 1669240325037,
        "lastModifiedDate": 1669240325037
    }
}
```

6. Buat kebijakan akses data yang memberikan izin minimum untuk mengindeks dan mencari data dalam koleksi buku. Ganti ARN utama dengan ARN dari `TutorialRole` langkah 1:

```
aws opensearchserverless create-access-policy \
  --name books-policy \
  --type data \
  --policy "[{\"Rules\":[{\"ResourceType\":\"index\",\"Resource\":
[\"index\/books\/books-index\"],\"Permission\":[\"aoss:CreateIndex
\",\"aoss:DescribeIndex\",\"aoss:ReadDocument\",\"aoss:WriteDocument
\",\"aoss:UpdateIndex\",\"aoss:DeleteIndex\"]}],\"Principal\":
[\"arn:aws:iam::123456789012:role\/TutorialRole\"]}]"
```

Sampel respon

```
{
    "accessPolicyDetail": {
        "type": "data",
        "name": "books-policy",
        "policyVersion": "MTY2OTI0MDM5NDY1M18x",
        "policy": [
            {
                "Rules": [
                    {
                        "Resource": [
                            "index/books/books-index"
                        ],
                        "Permission": [
                            "aoss:CreateIndex",
                            "aoss:DescribeIndex",
                            "aoss:ReadDocument",
                            "aoss:WriteDocument",
                            "aoss:UpdateDocument",
                            "aoss:DeleteDocument"
                        ],
                        "ResourceType": "index"
                    }
                ],
                "Principal": [
                    "arn:aws:iam::123456789012:role/TutorialRole"
                ]
            }
        ],
        "createdDate": 1669240394653,
        "lastModifiedDate": 1669240394653
    }
```

```
}
```

`TutorialRole`sekarang harus dapat mengindeks dan mencari dokumen dalam koleksi buku.

7. Untuk melakukan panggilan ke OpenSearch API, Anda memerlukan titik akhir koleksi. Kirim permintaan berikut untuk mengambil `collectionEndpoint` parameter:

```
aws opensearchserverless batch-get-collection --names books
```

Sampel respon

```
{
    "collectionDetails": [
        {
            "id": "8kw362bpwg4gx9b2f6e0",
            "name": "books",
            "status": "ACTIVE",
            "type": "SEARCH",
            "description": "",
            "arn": "arn:aws:aoss:us-
east-1:123456789012:collection/8kw362bpwg4gx9b2f6e0",
            "createdDate": 1665765327107,
            "collectionEndpoint": "https://8kw362bpwg4gx9b2f6e0.us-
east-1.aoss.amazonaws.com",
            "dashboardEndpoint": "https://8kw362bpwg4gx9b2f6e0.us-
east-1.aoss.amazonaws.com/_dashboards"
        }
    ],
    "collectionErrorDetails": []
}
```

> **Note**
>
> Anda tidak akan dapat melihat titik akhir koleksi hingga status koleksi berubah menjadi`ACTIVE`. Anda mungkin harus melakukan beberapa panggilan untuk memeriksa status hingga koleksi berhasil dibuat.

8. Gunakan alat HTTP seperti Postman atau curl untuk mengindeks data ke dalam koleksi buku. Kita akan membuat indeks yang disebut books-index dan menambahkan satu dokumen.

Kirim permintaan berikut ke titik akhir koleksi yang Anda ambil di langkah sebelumnya, menggunakan kredensialnya. `TutorialRole`

```
PUT https://8kw362bpwg4gx9b2f6e0.us-east-1.aoss.amazonaws.com/books-index/_doc/1
{
  "title": "The Shining",
  "author": "Stephen King",
  "year": 1977
}
```

Sampel respon

```
{
  "_index" : "books-index",
  "_id" : "1",
  "_version" : 1,
  "result" : "created",
  "_shards" : {
    "total" : 0,
    "successful" : 0,
    "failed" : 0
  },
  "_seq_no" : 0,
  "_primary_term" : 0
}
```

9. Untuk mulai mencari data dalam koleksi Anda, gunakan API pencarian. Kueri berikut melakukan pencarian dasar:

```
GET https://8kw362bpwg4gx9b2f6e0.us-east-1.aoss.amazonaws.com/books-index/_search
```

Sampel respon

```
{
    "took": 405,
    "timed_out": false,
    "_shards": {
        "total": 6,
        "successful": 6,
        "skipped": 0,
```

```
        "failed": 0
    },
    "hits": {
        "total": {
            "value": 2,
            "relation": "eq"
        },
        "max_score": 1.0,
        "hits": [
            {
                "_index": "books-index:0::3xJq14MBUaOS0wL26UU9:0",
                "_id": "F_bt4oMBLle5pYmm5q4T",
                "_score": 1.0,
                "_source": {
                    "title": "The Shining",
                    "author": "Stephen King",
                    "year": 1977
                }
            }
        ]
    }
}
```


Identity and Access Management untuk Amazon OpenSearch Serverless

(IAM) AWS Identity and Access Management adalah Layanan AWS yang membantu seorang administrator dalam mengendalikan akses ke sumber daya AWS secara aman. Administrator IAM mengontrol siapa yang dapat diautentikasi (masuk) dan diberi wewenang (memiliki izin) untuk menggunakan sumber daya Tanpa Server. OpenSearch IAM adalah sebuah layanan Layanan AWS yang dapat Anda gunakan tanpa dikenakan biaya tambahan.

Topik

- Kebijakan berbasis identitas untuk Tanpa Server OpenSearch
- Tindakan kebijakan untuk Tanpa OpenSearch Server
- Sumber daya kebijakan untuk Tanpa OpenSearch Server
- Kunci kondisi kebijakan untuk Amazon Tanpa OpenSearch Server
- ABAC dengan Tanpa Server OpenSearch
- Menggunakan kredensil sementara dengan Tanpa Server OpenSearch

- Peran terkait layanan untuk Tanpa Server OpenSearch
- Contoh kebijakan berbasis identitas untuk Tanpa Server OpenSearch

Kebijakan berbasis identitas untuk Tanpa Server OpenSearch

Mendukung kebijakan berbasis identitas	Ya

Kebijakan berbasis identitas adalah dokumen kebijakan izin JSON yang dapat Anda lampirkan ke sebuah identitas, misalnya pengguna IAM, grup pengguna IAM, atau peran IAM. Kebijakan ini mengontrol apa yang pengguna tindakan dan peran dapat kerjakan, pada sumber daya mana, dan dalam keadaan apa. Untuk mempelajari cara membuat kebijakan berbasis identitas, silakan lihat Membuat kebijakan IAM dalam Panduan Pengguna IAM.

Dengan kebijakan berbasis identitas IAM, Anda dapat menentukan secara spesifik apakah tindakan dan sumber daya diizinkan atau ditolak, serta persyaratan yang menjadi dasar dikabulkan atau ditolaknya tindakan tersebut. Anda tidak dapat menentukan secara spesifik pengguna utama dalam sebuah kebijakan berbasis identitas karena pengguna utama berlaku bagi pengguna atau peran yang melekat kepadanya. Untuk mempelajari semua elemen yang dapat Anda gunakan dalam kebijakan JSON, silakan lihat Referensi elemen kebijakan JSON IAM dalam Panduan Pengguna IAM.

Contoh kebijakan berbasis identitas untuk Tanpa Server OpenSearch

Untuk melihat contoh kebijakan berbasis identitas OpenSearch Tanpa Server, lihat. the section called “Contoh kebijakan berbasis identitas”

Tindakan kebijakan untuk Tanpa OpenSearch Server

Mendukung tindakan kebijakan	Ya

Elemen `Action` dari kebijakan JSON menjelaskan tindakan-tindakan yang dapat Anda gunakan untuk mengizinkan atau menolak akses dalam sebuah kebijakan. Tindakan-tindakan kebijakan biasanya memiliki nama yang sama sebagaimana operasi API AWS yang dikaitkan padanya. Ada beberapa pengecualian, misalnya tindakan yang memiliki izin saja yang tidak memiliki operasi API yang cocok. Ada juga beberapa operasi yang memerlukan beberapa tindakan dalam sebuah kebijakan. Tindakan-tindakan tambahan ini disebut tindakan dependen.

Menyertakan tindakan dalam suatu kebijakan untuk memberikan izin guna melakukan operasi yang terkait.

Tindakan kebijakan di OpenSearch Tanpa Server menggunakan awalan berikut sebelum tindakan:

```
aoss
```

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

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

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

```
"Action": "aoss:List*"
```

Untuk melihat contoh kebijakan berbasis identitas OpenSearch Tanpa Server, lihat. Contoh kebijakan berbasis identitas untuk Tanpa Server OpenSearch

Sumber daya kebijakan untuk Tanpa OpenSearch Server

Mendukung sumber daya kebijakan	Ya

Administrator dapat menggunakan kebijakan JSON AWS untuk menentukan secara spesifik siapa yang memiliki akses pada apa. Yaitu, pengguna utama manakah yang dapat melakukan tindakan pada sumber daya apa, dan dengan syarat apa.

Elemen kebijakan JSON `Resource` menentukan objek atau objek-objek yang menjadi target penerapan tindakan. Pernyataan harus menyertakan entah elemen `Resource` atau `NotResource`. Praktik terbaiknya, tentukan sumber daya menggunakan Amazon Resource Name (ARN). Anda dapat melakukan ini untuk tindakan-tindakan yang mendukung jenis sumber daya tertentu, yang dikenal sebagai izin tingkat sumber daya.

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

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

Kunci kondisi kebijakan untuk Amazon Tanpa OpenSearch Server

Mendukung kunci-kunci persyaratan kebijakan spesifik layanan	Ya

Administrator dapat menggunakan kebijakan JSON AWS untuk menentukan secara spesifik siapa yang memiliki akses pada apa. Yaitu, pengguna utama manakah yang dapat melakukan tindakan pada sumber daya apa, dan dengan syarat apa.

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

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

Anda juga dapat menggunakan variabel placeholder saat menentukan syarat. Sebagai contoh, Anda dapat memberikan izin kepada pengguna IAM untuk mengakses sumber daya hanya jika izin tersebut mempunyai tag yang sesuai dengan nama pengguna IAM mereka. Untuk informasi selengkapnya, silakan lihat Elemen kebijakan IAM: variabel dan tag di Panduan Pengguna IAM.

AWS mendukung kunci-kunci syarat global dan kunci-kunci syarat spesifik layanan. Untuk melihat semua kunci persyaratan global AWS, silakan lihat kunci konteks syarat global AWS di Panduan Pengguna IAM.

Selain kontrol akses berbasis atribut (ABAC), OpenSearch Tanpa Server mendukung kunci kondisi berikut:

- `aoss:collection`
- `aoss:CollectionId`
- `aoss:index`

Anda dapat menggunakan kunci kondisi ini bahkan ketika memberikan izin untuk kebijakan akses dan kebijakan keamanan. Sebagai contoh:

```
[
   {
      "Effect":"Allow",
      "Action":[
         "aoss:CreateAccessPolicy",
         "aoss:CreateSecurityPolicy"
      ],
      "Resource":"*",
      "Condition":{
         "StringLike":{
            "aoss:collection":"log"
         }
      }
   }
]
```

Dalam contoh ini, kondisi berlaku untuk kebijakan yang berisi aturan yang cocok dengan nama atau pola koleksi. Kondisi memiliki perilaku berikut:

- `StringEquals`- Berlaku untuk kebijakan dengan aturan yang berisi string sumber daya yang tepat "log" (yaitu`collection/log`).
- `StringLike`- Berlaku untuk kebijakan dengan aturan yang berisi string sumber daya yang menyertakan string "log" (yaitu `collection/log` tetapi juga `collection/logs-application` atau`collection/applogs123`).

> **Note**
>
> Kunci kondisi koleksi tidak berlaku di tingkat indeks. Misalnya, dalam kebijakan di atas, kondisi tidak akan berlaku untuk akses atau kebijakan keamanan yang berisi string sumber daya`index/logs-application/*`.

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

ABAC dengan Tanpa Server OpenSearch

Mendukung ABAC (tanda dalam kebijakan)	Ya

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

ABAC sangat berguna di lingkungan yang berkembang dengan cepat dan berguna di situasi dimana pengelolaan kebijakan menjadi rumit.

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

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

Untuk informasi selengkapnya tentang ABAC, silakan lihat Apa itu ABAC? di Panduan Pengguna IAM. Untuk melihat tutorial yang menguraikan langkah-langkah pengaturan ABAC, silakan lihat Menggunakan kontrol akses berbasis atribut (ABAC) di Panduan Pengguna IAM.

Untuk informasi selengkapnya tentang menandai sumber daya OpenSearch Tanpa Server, lihat. the section called “Penandaan koleksi”

Menggunakan kredensil sementara dengan Tanpa Server OpenSearch

Mendukung kredensial temporer	Ya

Beberapa Layanan AWS tidak berfungsi saat Anda masuk dengan menggunakan kredensial temporer. Sebagai informasi tambahan, termasuk tentang Layanan AWS mana saja yang berfungsi dengan kredensial temporer, silakan lihat Layanan AWS yang berfungsi dengan IAM di Panduan Pengguna IAM.

Anda menggunakan kredensial temporer jika Anda masuk ke AWS Management Console dengan menggunakan metode apa pun kecuali nama pengguna dan kata sandi. Sebagai contoh, ketika Anda mengakses AWS dengan menggunakan tautan masuk tunggal (SSO) milik perusahaan Anda, proses itu secara otomatis akan membuat kredensial temporer. Anda juga akan secara otomatis membuat kredensial temporer ketika Anda masuk ke konsol sebagai seorang pengguna dan kemudian beralih peran. Untuk informasi selengkapnya tentang peralihan peran, silakan lihat Peralihan peran (konsol) di Panduan Pengguna IAM.

Anda dapat secara manual membuat kredensial temporer menggunakan AWS CLI atau API AWS. Anda kemudian dapat menggunakan kredensial temporer tersebut untuk mengakses AWS. AWS menyarankan agar Anda secara dinamis membuat kredensial temporer alih-alih menggunakan kunci akses jangka panjang. Untuk informasi selengkapnya, silakan lihat Kredensial keamanan temporer di IAM.

Peran terkait layanan untuk Tanpa Server OpenSearch

Mendukung peran yang terhubung dengan layanan	Ya

Peran yang tertaut layanan adalah jenis peran layanan yang tertaut dengan Layanan AWS. Layanan tersebut dapat menjalankan peran untuk melakukan sebuah tindakan atas nama Anda. Peran tertaut layanan akan muncul di Akun AWS Anda dan dimiliki oleh layanan tersebut. Administrator IAM dapat melihat, tetapi tidak dapat mengedit izin untuk peran tertaut layanan.

Untuk detail tentang membuat dan mengelola peran terkait layanan OpenSearch Tanpa Server, lihat. the section called “Peran pembuatan koleksi”

Contoh kebijakan berbasis identitas untuk Tanpa Server OpenSearch

Secara default, pengguna dan peran tidak memiliki izin untuk membuat atau memodifikasi sumber daya OpenSearch Tanpa Server. Pengguan dan peran tersebut juga tidak dapat melakukan tugas dengan menggunakan API AWS Management Console, AWS Command Line Interface (AWS CLI),

atau AWS. Untuk mengabulkan izin kepada pengguna untuk melakukan tindakan di sumber daya yang mereka perlukan, administrator IAM dapat membuat kebijakan IAM. Administrator kemudian akan dapat menambahkan kebijakan IAM ke peran, dan para pengguna dapat mengambil peran.

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

Untuk detail tentang tindakan dan jenis sumber daya yang ditentukan oleh Amazon OpenSearch Tanpa Server, termasuk format ARN untuk setiap jenis sumber daya, lihat Tindakan, sumber daya, dan kunci kondisi untuk Amazon OpenSearch Tanpa Server di Referensi Otorisasi Layanan.

Topik

- Praktik terbaik kebijakan
- Menggunakan OpenSearch Tanpa Server di konsol
- Mengelola koleksi OpenSearch Tanpa Server
- Melihat OpenSearch koleksi Tanpa Server
- Menggunakan operasi OpenSearch API

Praktik terbaik kebijakan

Kebijakan berbasis identitas adalah pilihan yang sangat tepat. Mereka menentukan apakah seseorang dapat membuat, mengakses, atau menghapus sumber daya OpenSearch Tanpa Server di akun Anda. Tindakan ini mengenakan biaya kepada Anda Akun AWS. Ketika Anda membuat atau mengedit kebijakan berbasis identitas, ikuti panduan dan rekomendasi ini:

Kebijakan berbasis identitas menentukan apakah seseorang dapat membuat, mengakses, atau menghapus sumber daya OpenSearch Tanpa Server di akun Anda. Tindakan ini mengenakan biaya kepada Anda Akun AWS. Ketika Anda membuat atau mengedit kebijakan berbasis identitas, ikuti panduan dan rekomendasi ini:

- Mulailah dengan kebijakan terkelola AWS dan beralih ke izin dengan hak akses paling rendah – Untuk mulai memberikan izin kepada pengguna dan beban kerja Anda, gunakan kebijakan terkelola AWS yang memberikan izin untuk banyak kasus penggunaan umum. Kebijakan tedapat di Akun AWS Anda. Kami menyarankan Anda untuk mengurangi izin lebih lanjut dengan menentukan kebijakan yang dikelola pelanggan AWS yang khusus untuk kasus penggunaan Anda. Untuk informasi selengkapnya, silakan lihat kebijakan-kebijakan terkelola AWS atau kebijakan-kebijakan terkelola AWS untuk fungsi tugas di Panduan Pengguna IAM.

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

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

Menggunakan OpenSearch Tanpa Server di konsol

Untuk mengakses OpenSearch Tanpa Server dalam konsol OpenSearch Layanan, Anda harus memiliki set izin minimum. Izin ini harus memungkinkan Anda untuk membuat daftar dan melihat detail tentang sumber daya OpenSearch Tanpa Server di akun Anda. AWS Jika Anda membuat kebijakan berbasis identitas yang lebih ketat daripada izin minimum yang diperlukan, konsol tidak akan berfungsi sebagaimana dimaksud untuk entitas (seperti peran IAM) dengan kebijakan tersebut.

Anda tidak perlu mengizinkan konsol minimum untuk pengguna yang melakukan panggilan hanya ke AWS CLI atau API AWS. Alih-alih, izinkan akses hanya ke tindakan yang sesuai dengan operasi API yang Anda coba lakukan.

Kebijakan berikut memungkinkan pengguna mengakses OpenSearch Tanpa Server dalam konsol OpenSearch Layanan:

```
{
    "Version": "2012-10-17",
    "Statement": [
        {
            "Resource": "*",
            "Effect": "Allow",
            "Action": [
                "aoss:ListCollections",
                "aoss:BatchGetCollection",
                "aoss:ListAccessPolicies",
                "aoss:ListSecurityConfigs",
                "aoss:ListSecurityPolicies",
                "aoss:ListTagsForResource",
                "aoss:ListVpcEndpoints",
                "aoss:GetAccessPolicy",
                "aoss:GetAccountSettings",
                "aoss:GetSecurityConfig",
                "aoss:GetSecurityPolicy"
            ]
        }
    ]
}
```

Mengelola koleksi OpenSearch Tanpa Server

Kebijakan ini adalah contoh kebijakan “admin koleksi” yang memungkinkan pengguna mengelola dan mengelola koleksi Amazon Tanpa OpenSearch Server. Pengguna dapat membuat, melihat, dan menghapus koleksi.

```
{
    "Version": "2012-10-17",
    "Statement": [
        {
            "Resource": "arn:aws:aoss:region:123456789012:collection/*",
            "Action": [
```

```
                "aoss:CreateCollection",
                "aoss:DeleteCollection",
                "aoss:UpdateCollection"
            ],
            "Effect": "Allow"
        },
        {
            "Resource": "*",
            "Action": [
                "aoss:BatchGetCollection",
                "aoss:ListCollections",
                "aoss:CreateAccessPolicy",
                "aoss:CreateSecurityPolicy"
            ],
            "Effect": "Allow"
        }
    ]
}
```

Melihat OpenSearch koleksi Tanpa Server

Kebijakan contoh ini memungkinkan pengguna untuk melihat detail untuk semua koleksi Amazon OpenSearch Tanpa Server di akun mereka. Pengguna tidak dapat mengubah koleksi atau kebijakan keamanan terkait.

```
{
    "Version": "2012-10-17",
    "Statement": [
        {
            "Resource": "*",
            "Action": [
                "aoss:ListAccessPolicies",
                "aoss:ListCollections",
                "aoss:ListSecurityPolicies",
                "aoss:ListTagsForResource",
                "aoss:BatchGetCollection"
            ],
            "Effect": "Allow"
        }
    ]
}
```

Menggunakan operasi OpenSearch API

Operasi API bidang data terdiri dari fungsi yang Anda gunakan di OpenSearch Tanpa Server untuk mendapatkan nilai realtime dari layanan. Operasi API bidang kontrol terdiri dari fungsi yang Anda gunakan untuk mengatur lingkungan.

Untuk mengakses API dan OpenSearch Dasbor bidang data Amazon OpenSearch Tanpa Server dari browser, Anda perlu menambahkan dua izin IAM untuk sumber daya pengumpulan. Izin ini adalah `aoss:APIAccessAll` dan`aoss:DashboardsAccessAll`.

> **Note**
>
> Mulai 10 Mei 2023, OpenSearch Tanpa Server memerlukan dua izin IAM baru ini untuk sumber daya pengumpulan. `aoss:APIAccessAll`Izin memungkinkan akses pesawat data, dan `aoss:DashboardsAccessAll` izin memungkinkan OpenSearch Dasbor dari browser. Kegagalan untuk menambahkan dua izin IAM baru menghasilkan kesalahan 403.

Kebijakan contoh ini memungkinkan pengguna mengakses API bidang data untuk koleksi tertentu di akun mereka, dan mengakses OpenSearch Dasbor untuk semua koleksi di akun mereka.

```
{
    "Version": "2012-10-17",
    "Statement": [
         {
            "Effect": "Allow",
            "Action": "aoss:APIAccessAll",
            "Resource": "arn:aws:aoss:region:account-id:collection/collection-id"
        },
        {
            "Effect": "Allow",
            "Action": "aoss:DashboardsAccessAll",
            "Resource": "arn:aws:aoss:region:account-id:dashboards/default"
        }
    ]
}
```

Keduanya `aoss:APIAccessAll` dan `aoss:DashboardsAccessAll` memberikan izin IAM penuh ke sumber daya pengumpulan, sementara izin Dasbor juga menyediakan akses OpenSearch Dasbor. Setiap izin bekerja secara independen, jadi penolakan eksplisit `aoss:APIAccessAll` tidak

memblokir `aoss:DashboardsAccessAll` akses ke sumber daya, termasuk Alat Pengembang. Hal yang sama berlaku untuk penyangkalan`aoss:DashboardsAccessAll`.

OpenSearch Tanpa server hanya mendukung alamat IP sumber dalam pengaturan kondisi dalam kebijakan IAM kepala sekolah untuk panggilan pesawat data:

```
"Condition": {
    "IpAddress": {
         "aws:SourceIp": "52.95.4.14"
    }
}
```

Enkripsi di Amazon OpenSearch Tanpa Server

Enkripsi saat tidak aktif

Setiap koleksi Amazon OpenSearch Tanpa Server yang Anda buat dilindungi dengan enkripsi data saat tidak digunakan, fitur keamanan yang membantu mencegah akses yang tidak sah ke data Anda. Enkripsi saat istirahat menggunakanAWS Key Management Service (AWS KMS) untuk menyimpan dan mengelola kunci enkripsi Anda. Menggunakan algoritma Advanced Encryption Standard dengan kunci 256-bit (AES-256) untuk melakukan enkripsi.

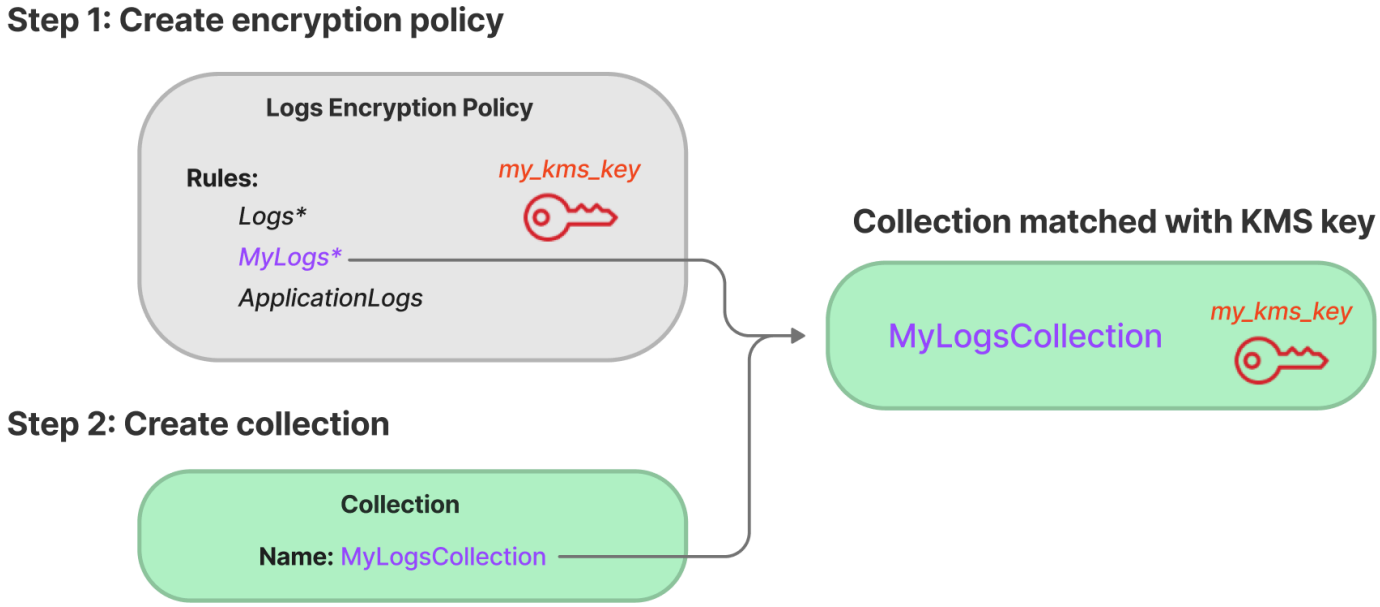
Topik

- Kebijakan enkripsi
- Pertimbangan-pertimbangan
- Izin diperlukan
- Kebijakan utama untuk kunci yang dikelola pelanggan
- Bagaimana OpenSearch Serverless menggunakan hibah diAWS KMS
- Membuat kebijakan enkripsi (konsol)
- Membuat kebijakan enkripsi (AWS CLI)
- Melihat kebijakan enkripsi
- Memperbarui kebijakan enkripsi
- Menghapus kebijakan enkripsi

Kebijakan enkripsi

Dengan kebijakan enkripsi, Anda dapat mengelola banyak koleksi dalam skala besar dengan secara otomatis menetapkan kunci enkripsi ke koleksi yang baru dibuat yang cocok dengan nama atau pola tertentu.

Saat membuat kebijakan enkripsi, Anda dapat menentukan awalan, yang merupakan aturan pencocokan berbasis wildcard sepertiMyCollection*, atau memasukkan satu nama koleksi. Kemudian, ketika Anda membuat koleksi yang cocok dengan nama atau pola awalan, kebijakan dan kunci KMS terkait secara otomatis ditetapkan untuk itu.



Kebijakan enkripsi berisi elemen berikut:

- Rules- satu atau lebih aturan pencocokan koleksi, masing-masing dengan sub-elemen berikut:
  - ResourceType— Saat ini satu-satunya pilihan adalah “koleksi”. Kebijakan enkripsi hanya berlaku untuk pengumpulan sumber daya.
  - Resource- Satu atau lebih nama koleksi atau pola yang akan diterapkan kebijakan, dalam formatcollection/<*collection name|pattern*>.
- AWSOwnedKey- Apakah akan menggunakanKunci milik AWS.
- KmsARN— Jika Anda menyetelAWSOwnedKey ke false, tentukan Amazon Resource Name (ARN) dari kunci KMS untuk mengenkripsi koleksi yang terkait. Jika Anda menyertakan parameter ini, OpenSearch Tanpa Server mengabaikanAWSOwnedKey parameter.

Kebijakan sampel berikut akan menetapkan kunci yang dikelola pelanggan untuk koleksi future yang disebutkan`autopartsinventory`, serta koleksi yang dimulai dengan istilah “penjualan”:

```
{
   "Rules":[
      {
         "ResourceType":"collection",
         "Resource":[
            "collection/autopartsinventory",
            "collection/sales*"
         ]
      }
   ],
   "AWSOwnedKey":false,
   "KmsARN":"arn:aws:encryption:us-east-1:123456789012:key/93fd6da4-a317-4c17-
bfe9-382b5d988b36"
}
```

Meskipun kebijakan cocok dengan nama koleksi, Anda dapat memilih untuk mengganti penetapan otomatis ini selama pembuatan koleksi jika pola sumber daya berisi wildcard (*). Jika Anda memilih untuk mengganti penetapan kunci otomatis, OpenSearch Tanpa Server membuat kebijakan enkripsi untuk Anda bernama auto-**< *collection-name* >** dan melampirkannya ke koleksi. Kebijakan awalnya hanya berlaku untuk satu koleksi, tetapi Anda dapat memodifikasinya untuk menyertakan koleksi tambahan.

Jika Anda mengubah aturan kebijakan agar tidak lagi cocok dengan koleksi, kunci KMS terkait tidak akan ditetapkan dari koleksi itu. Koleksi selalu tetap dienkripsi dengan kunci enkripsi awalnya. Jika Anda ingin mengubah kunci enkripsi untuk koleksi, Anda harus membuat ulang koleksi.

Jika aturan dari beberapa kebijakan cocok dengan koleksi, aturan yang lebih spesifik akan digunakan. Misalnya, jika satu kebijakan berisi aturan untuk`collection/log*`, dan lainnya untuk`collection/logSpecial`, kunci enkripsi untuk kebijakan kedua digunakan karena lebih spesifik.

Anda tidak dapat menggunakan nama atau awalan dalam kebijakan jika sudah ada di kebijakan lain. OpenSearch Tanpa server menampilkan kesalahan jika Anda mencoba mengonfigurasi pola sumber daya yang identik dalam kebijakan enkripsi yang berbeda.

Pertimbangan-pertimbangan

Pertimbangkan hal berikut saat Anda mengkonfigurasi enkripsi untuk koleksi Anda:

- Enkripsi saat istirahat diperlukan untuk semua koleksi tanpa server.
- Anda memiliki opsi untuk menggunakan kunci yang dikelola pelanggan atauKunci milik AWS. Jika Anda memilih kunci yang dikelola pelanggan, kami sarankan Anda mengaktifkan rotasi kunci otomatis.
- Anda tidak dapat mengubah kunci enkripsi untuk koleksi setelah koleksi dibuat. Hati-hatiAWS KMS memilih mana yang akan digunakan pertama kali Anda mengatur koleksi.
- Koleksi hanya dapat mencocokkan kebijakan enkripsi tunggal.
- Koleksi dengan kunci KMS unik tidak dapat berbagi OpenSearch Compute Units (OCU) dengan koleksi lainnya. Setiap koleksi dengan kunci unik membutuhkan 4 OCU sendiri.
- Jika Anda memperbarui kunci KMS dalam kebijakan enkripsi, perubahan tersebut tidak memengaruhi koleksi yang cocok dengan kunci KMS yang sudah ditetapkan.
- OpenSearch Tanpa server tidak secara eksplisit memeriksa izin pengguna pada kunci yang dikelola pelanggan. Jika pengguna memiliki izin untuk mengakses koleksi melalui kebijakan akses data, mereka akan dapat menelan dan meminta data yang dienkripsi dengan kunci terkait.

Izin diperlukan

Enkripsi saat istirahat untuk OpenSearch Tanpa Server menggunakan izinAWS Identity and Access Management (IAM) berikut. Anda dapat menentukan kondisi IAM untuk membatasi pengguna pada koleksi tertentu.

- `aoss:CreateSecurityPolicy`- Buat kebijakan enkripsi.
- `aoss:ListSecurityPolicies`- Cantumkan semua kebijakan enkripsi dan koleksi yang dilampirkan.
- `aoss:GetSecurityPolicy`- Lihat detail kebijakan enkripsi tertentu.
- `aoss:UpdateSecurityPolicy`- Memodifikasi kebijakan enkripsi.
- `aoss:DeleteSecurityPolicy`- Hapus kebijakan enkripsi.

Contoh kebijakan akses berbasis identitas berikut menyediakan izin minimum yang diperlukan bagi pengguna untuk mengelola kebijakan enkripsi dengan pola sumber daya`collection/application-logs`.

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

```
      {
         "Effect":"Allow",
         "Action":[
            "aoss:CreateSecurityPolicy",
            "aoss:UpdateSecurityPolicy",
            "aoss:DeleteSecurityPolicy",
            "aoss:GetSecurityPolicy"
         ],
         "Resource":"*",
         "Condition":{
            "StringEquals":{
               "aoss:collection":"application-logs"
            }
         }
      },
      {
         "Effect":"Allow",
         "Action":[
            "aoss:ListSecurityPolicies"
         ],
         "Resource":"*"
      }
   ]
}
```

Kebijakan utama untuk kunci yang dikelola pelanggan

Jika Anda memilih kunci yang dikelola pelanggan untuk melindungi koleksi, OpenSearch Tanpa Server mendapat izin untuk menggunakan kunci KMS atas nama prinsipal yang membuat pilihan. Prinsipal tersebut, pengguna atau peran, harus memiliki izin pada kunci KMS yang diperlukan OpenSearch Tanpa Server. Anda dapat memberikan izin ini di kebijakan kunci atau IAM.

Minimal, OpenSearch Tanpa Server memerlukan izin berikut pada kunci yang dikelola pelanggan:

- km:DescribeKey
- km:CreateGrant
- km:ListKeys

Misalnya:

```
{
```

```
  "Version": "2012-10-17",
  "Statement": [
    {
      "Effect": "Allow",
      "Action": [
        "kms:ListKeys"
      ],
      "Resource": "*"
    },
{
      "Effect": "Allow",
      "Action": [
        "kms:DescribeKey",
        "kms:CreateGrant"
      ],
      "Resource": "{kms-key-arn}"
    }
  ]
}
```

OpenSearch Tanpa server membuat hibah dengan kms:GenerateDataKey dan kms:Decrypt perizinan.

Jika Anda ingin menjaga kunci eksklusif Anda untuk OpenSearch Tanpa Server, Anda dapat menambahkanViaService kondisi km: untuk kebijakan kunci tersebut:

```
"Condition": {
  "StringEquals": {
    "kms:ViaService": "aoss.us-east-1.amazonaws.com"
  },
  "Bool": {
    "kms:GrantIsForAWSResource": "true"
  }
}
```

Untuk informasi selengkapnya, lihat Menggunakan kebijakan kunci di AWS KMS di Panduan Developer AWS Key Management Service.

Bagaimana OpenSearch Serverless menggunakan hibah diAWS KMS

OpenSearch Tanpa server memerlukan hibah untuk menggunakan kunci yang dikelola pelanggan.

Saat Anda membuat kebijakan enkripsi di akun Anda dengan kunci baru, OpenSearch Tanpa Server membuat hibah atas nama Anda dengan mengirimkan CreateGrantpermintaanAWS KMS. HibahAWS KMS digunakan untuk memberikan akses OpenSearch Tanpa Server ke kunci KMS di akun pelanggan.

OpenSearch Tanpa server memerlukan hibah untuk menggunakan kunci yang dikelola pelanggan Anda untuk operasi internal berikut:

- Kirim DescribeKeypermintaanAWS KMS untuk memverifikasi bahwa ID kunci yang dikelola pelanggan simetris yang diberikan valid.
- Kirim GenerateDataKeypermintaan ke kunci KMS untuk membuat kunci data yang dapat digunakan untuk mengenkripsi objek.
- Kirim permintaan DekripsiAWS KMS untuk mendekripsi kunci data terenkripsi sehingga mereka dapat digunakan untuk mengenkripsi data Anda.

Anda dapat mencabut akses ke hibah, atau menghapus akses layanan ke kunci yang dikelola pelanggan kapan saja. Jika Anda melakukannya, OpenSearch Tanpa Server tidak akan dapat mengakses data apa pun yang dienkripsi oleh kunci yang dikelola pelanggan, yang memengaruhi semua operasi yang bergantung pada data tersebut, yang menyebabkan`AccessDeniedException` kesalahan dan kegagalan dalam alur kerja asinkron.

OpenSearch Tanpa server menghentikan hibah dalam alur kerja asinkron saat kunci terkelola pelanggan tertentu tidak terkait dengan kebijakan atau koleksi keamanan apa pun.

Membuat kebijakan enkripsi (konsol)

Dalam kebijakan enkripsi, Anda menentukan kunci KMS dan serangkaian pola pengumpulan yang akan diterapkan kebijakan. Setiap koleksi baru yang cocok dengan salah satu pola yang ditentukan dalam kebijakan akan ditetapkan kunci KMS yang sesuai saat Anda membuat koleksi. Sebaiknya buat kebijakan enkripsi sebelum mulai membuat koleksi.

Membuat kebijakan enkripsi OpenSearch Tanpa Server

1. Buka konsol OpenSearch Layanan Amazon di https://console.aws.amazon.com/aos/home.
2. Di panel navigasi kiri, luaskan Tanpa Server dan pilih Kebijakan enkripsi.
3. Pilih Buat kebijakan enkripsi.
4. Berikan nama dan deskripsi untuk kebijakan.

5. Di bawah Sumber Daya, masukkan satu atau beberapa pola sumber daya untuk kebijakan enkripsi ini. Setiap koleksi yang baru dibuat di saat iniAkun AWS dan Wilayah yang cocok dengan salah satu pola secara otomatis ditetapkan ke kebijakan ini. Misalnya, jika Anda memasukkan`ApplicationLogs` (tanpa wildcard), dan kemudian membuat koleksi dengan nama itu, kebijakan dan kunci KMS terkait ditetapkan ke koleksi itu.

   Anda juga dapat memberikan awalan seperti`Logs*`, yang menetapkan kebijakan untuk koleksi baru dengan nama dimulai dengan`Logs`. Dengan menggunakan wildcard, Anda dapat mengelola pengaturan enkripsi untuk beberapa koleksi dalam skala besar.
6. Di bawah Enkripsi, pilih kunci KMS untuk digunakan.
7. Pilih Create (Buat).

Langkah berikutnya: Buat koleksi

Setelah mengonfigurasi satu atau beberapa kebijakan enkripsi, Anda dapat mulai membuat koleksi yang sesuai dengan aturan yang ditetapkan dalam kebijakan tersebut. Untuk petunjuk, lihat the section called “Membuat koleksi”.

Dalam langkah Enkripsi pembuatan koleksi, OpenSearch Tanpa Server memberi tahu Anda bahwa nama yang Anda masukkan cocok dengan pola yang ditentukan dalam kebijakan enkripsi, dan secara otomatis menetapkan kunci KMS yang sesuai ke koleksi. Jika pola sumber daya berisi wildcard (*), Anda dapat memilih untuk mengganti kecocokan dan memilih kunci Anda sendiri.

Membuat kebijakan enkripsi (AWS CLI)

Untuk membuat kebijakan enkripsi menggunakan operasi API OpenSearch Tanpa Server, Anda menentukan pola sumber daya dan kunci enkripsi dalam format JSON. CreateSecurityPolicyPermintaan menerima kebijakan inline dan file.json.

Kebijakan enkripsi memiliki format berikut ini. `my-policy.json`File sampel ini cocok dengan koleksi future yang dinamai`autopartsinventory`, serta koleksi apa pun dengan nama yang diawali`sales`.

```
{
   "Rules":[
      {
         "ResourceType":"collection",
         "Resource":[
            "collection/autopartsinventory",
```

```
            "collection/sales*"
         ]
      }
   ],
   "AWSOwnedKey":false,
   "KmsARN":"arn:aws:encryption:us-east-1:123456789012:key/93fd6da4-a317-4c17-
bfe9-382b5d988b36"
}
```

Untuk menggunakan kunci milik layanan, aturAWSOwnedKey ketrue:

```
{
   "Rules":[
      {
         "ResourceType":"collection",
         "Resource":[
            "collection/autopartsinventory",
            "collection/sales*"
         ]
      }
   ],
   "AWSOwnedKey":true
}
```

Permintaan berikut membuat kebijakan enkripsi:

```
aws opensearchserverless create-security-policy \
    --name sales-inventory \
    --type encryption \
    --policy file://my-policy.json
```

Kemudian, gunakan operasi CreateCollectionAPI untuk membuat satu atau lebih koleksi yang cocok dengan salah satu pola sumber daya.

Melihat kebijakan enkripsi

Sebelum membuat koleksi, Anda mungkin ingin melihat pratinjau kebijakan enkripsi yang ada di akun Anda untuk melihat mana yang memiliki pola sumber daya yang cocok dengan nama koleksi Anda. ListSecurityPoliciesPermintaan berikut mencantumkan semua kebijakan enkripsi di akun Anda:

```
aws opensearchserverless list-security-policies --type encryption
```

Permintaan mengembalikan informasi tentang semua kebijakan enkripsi yang dikonfigurasi. Gunakan isi`policy` elemen untuk melihat aturan pola yang didefinisikan dalam kebijakan:

```
{
   "securityPolicyDetails": [
      {
         "createdDate": 1663693217826,
         "description": "Sample encryption policy",
         "lastModifiedDate": 1663693217826,
         "name": "my-policy",
         "policy": "{\"Rules\":[{\"ResourceType\":\"collection\",\"Resource\":
[\"collection/autopartsinventory\",\"collection/sales*\"]}],\"AWSOwnedKey\":true}",
         "policyVersion": "MTY2MzY5MzIxNzgyNl8x",
         "type": "encryption"
      }
   ]
}
```

Untuk melihat informasi terperinci tentang kebijakan tertentu, termasuk kunci KMS, gunakan GetSecurityPolicyperintah.

Memperbarui kebijakan enkripsi

Jika Anda memperbarui kunci KMS dalam kebijakan enkripsi, perubahan hanya berlaku untuk koleksi yang baru dibuat yang cocok dengan nama atau pola yang dikonfigurasi. Ini tidak mempengaruhi koleksi yang ada yang memiliki kunci KMS sudah ditetapkan.

Hal yang sama berlaku untuk aturan pencocokan kebijakan. Jika Anda menambahkan, memodifikasi, atau menghapus aturan, perubahan hanya berlaku untuk koleksi yang baru dibuat. Koleksi yang ada tidak akan kehilangan kunci KMS yang ditetapkan jika Anda mengubah aturan kebijakan sehingga tidak lagi cocok dengan nama koleksi.

Untuk memperbarui kebijakan enkripsi di konsol OpenSearch Tanpa Server, pilih Kebijakan enkripsi, pilih kebijakan yang akan diubah, dan pilih Edit. Buat perubahan dan pilih Simpan.

Untuk memperbarui kebijakan enkripsi menggunakan API OpenSearch Tanpa Server, gunakan UpdateSecurityPolicyoperasi. Permintaan berikut memperbarui kebijakan enkripsi dengan dokumen JSON kebijakan baru:

```
aws opensearchserverless update-security-policy \
    --name sales-inventory \
    --type encryption \
```

```
    --policy-version 2 \
    --policy file://my-new-policy.json
```

Menghapus kebijakan enkripsi

Saat Anda menghapus kebijakan enkripsi, koleksi apa pun yang saat ini menggunakan kunci KMS yang ditentukan dalam kebijakan tidak akan terpengaruh. Untuk menghapus kebijakan di konsol OpenSearch Tanpa Server, pilih kebijakan dan pilih Hapus.

Anda juga dapat menggunakan DeleteSecurityPolicyoperasi:

```
aws opensearchserverless delete-security-policy --name my-policy --type encryption
```

Enkripsi dalam transit

Dalam OpenSearch Tanpa Server, semua jalur dalam koleksi dienkripsi saat transit menggunakan Transport Layer Security 1.2 (TLS) dengan cipher AES-256 standar industri. Akses ke semua API dan Dasbor untuk Opensearch juga melalui TLS 1.2. TLS adalah seperangkat protokol kriptografi standar industri yang digunakan untuk mengenkripsi informasi yang dipertukarkan melalui jaringan.

Akses jaringan untuk Amazon Tanpa OpenSearch Server

Pengaturan jaringan untuk koleksi Amazon OpenSearch Tanpa Server menentukan apakah koleksi dapat diakses melalui internet dari jaringan publik, atau apakah harus diakses secara pribadi.

Akses pribadi dapat berlaku untuk salah satu atau kedua hal berikut:

- OpenSearch Titik akhir VPC yang dikelola tanpa server
- Didukung Layanan AWS seperti Amazon Bedrock

Anda dapat mengonfigurasi akses jaringan secara terpisah untuk OpenSearchtitik akhir koleksi dan titik akhir OpenSearch Dasbor yang sesuai.

Akses jaringan adalah mekanisme isolasi untuk memungkinkan akses dari jaringan sumber yang berbeda. Misalnya, jika titik akhir OpenSearch Dasbor koleksi dapat diakses publik tetapi titik akhir OpenSearch API tidak, pengguna dapat mengakses data pengumpulan hanya melalui Dasbor saat menghubungkan dari jaringan publik. Jika mereka mencoba memanggil OpenSearch API langsung dari jaringan publik, mereka akan diblokir. Pengaturan jaringan dapat digunakan untuk permutasi sumber ke jenis sumber daya.

Topik

Kebijakan jaringan

Kebijakan jaringan memungkinkan Anda mengelola banyak koleksi dalam skala besar dengan secara otomatis menetapkan setelan akses jaringan ke koleksi yang cocok dengan aturan yang ditentukan dalam kebijakan.

Dalam kebijakan jaringan, Anda menentukan serangkaian aturan. Aturan ini menentukan izin akses ke titik akhir koleksi dan titik akhir OpenSearch Dasbor. Setiap aturan terdiri dari jenis akses (publik atau pribadi) dan jenis sumber daya (koleksi dan/atau titik akhir OpenSearch Dasbor). Untuk setiap jenis sumber daya (`collection`dan`dashboard`), Anda menentukan serangkaian aturan yang menentukan koleksi mana kebijakan akan diterapkan.

Dalam kebijakan contoh ini, aturan pertama menentukan akses titik akhir VPC ke titik akhir koleksi dan titik akhir Dasbor untuk semua koleksi yang dimulai dengan istilah. `marketing*` Ini juga menentukan akses Amazon Bedrock.

> **Note**
>
> Akses pribadi ke Layanan AWS seperti Amazon Bedrock hanya berlaku untuk titik akhir koleksi, bukan ke OpenSearch titik akhir OpenSearch Dasbor. Bahkan jika `ResourceType` ada`dashboard`, Layanan AWS tidak dapat diberikan akses ke OpenSearch Dasbor.

Aturan kedua menentukan akses publik ke `finance` koleksi, tetapi hanya untuk titik akhir koleksi (tidak ada akses Dasbor).

```
[
   {
      "Description":"Marketing access",
      "Rules":[
         {
            "ResourceType":"collection",
            "Resource":[
               "collection/marketing*"
            ]
         },
         {
            "ResourceType":"dashboard",
            "Resource":[
               "collection/marketing*"
            ]
         }
      ],
      "AllowFromPublic":false,
      "SourceVPCEs":[
         "vpce-050f79086ee71ac05"
      ],
      "SourceServices":[
         "bedrock.amazonaws.com"
      ],
   },
   {
      "Description":"Sales access",
      "Rules":[
         {
            "ResourceType":"collection",
            "Resource":[
               "collection/finance"
            ]
         }
      ],
      "AllowFromPublic":true
   }
]
```

Kebijakan ini menyediakan akses publik hanya ke OpenSearch Dasbor untuk koleksi yang dimulai dengan “keuangan”. Setiap upaya untuk mengakses OpenSearch API secara langsung akan gagal.

```
[
```

```
  {
    "Description": "Dashboards access",
    "Rules": [
      {
        "ResourceType": "dashboard",
        "Resource": [
          "collection/finance*"
        ]
      }
    ],
    "AllowFromPublic": true
  }
]
```

Kebijakan jaringan dapat berlaku untuk koleksi yang ada serta koleksi masa depan. Misalnya, Anda dapat membuat koleksi dan kemudian membuat kebijakan jaringan dengan aturan yang cocok dengan nama koleksi. Anda tidak perlu membuat kebijakan jaringan sebelum membuat koleksi.

Pertimbangan

Pertimbangkan hal berikut saat Anda mengonfigurasi akses jaringan untuk koleksi Anda:

- Jika Anda berencana untuk mengonfigurasi akses titik akhir VPC untuk koleksi, Anda harus terlebih dahulu membuat setidaknya satu titik akhir VPC yang dikelola Tanpa ServerOpenSearch .
- Akses pribadi Layanan AWS hanya berlaku untuk OpenSearch titik akhir koleksi, bukan ke titik akhir OpenSearch Dasbor. Bahkan jika `ResourceType` ada`dashboard`, Layanan AWS tidak dapat diberikan akses ke OpenSearch Dasbor.
- Jika koleksi dapat diakses dari jaringan publik, koleksi ini juga dapat diakses dari semua titik akhir VPC yang OpenSearch dikelola tanpa server dan semuanya. Layanan AWS
- Beberapa kebijakan jaringan dapat berlaku untuk satu koleksi. Untuk informasi selengkapnya, lihat the section called “Prioritas kebijakan”.

Izin diperlukan

Akses jaringan untuk OpenSearch Tanpa Server menggunakan izin AWS Identity and Access Management (IAM) berikut. Anda dapat menentukan kondisi IAM untuk membatasi pengguna pada kebijakan jaringan yang terkait dengan koleksi tertentu.

- `aoss:CreateSecurityPolicy`— Buat kebijakan akses jaringan.

- `aoss:ListSecurityPolicies`— Buat daftar semua kebijakan jaringan di akun saat ini.
- `aoss:GetSecurityPolicy`— Lihat spesifikasi kebijakan akses jaringan.
- `aoss:UpdateSecurityPolicy`— Ubah kebijakan akses jaringan tertentu, dan ubah ID VPC atau penunjukan akses publik.
- `aoss:DeleteSecurityPolicy`— Hapus kebijakan akses jaringan (setelah terlepas dari semua koleksi).

Kebijakan akses berbasis identitas berikut memungkinkan pengguna untuk melihat semua kebijakan jaringan, dan memperbarui kebijakan dengan pola sumber daya: `collection/application-logs`

```
{
    "Version": "2012-10-17",
    "Statement": [
        {
            "Effect": "Allow",
            "Action": [
                "aoss:UpdateSecurityPolicy"
            ],
            "Resource": "*",
            "Condition": {
                "StringEquals": {
                    "aoss:collection": "application-logs"
                }
            }
        },
        {
            "Effect": "Allow",
            "Action": [
                "aoss:ListSecurityPolicies",
                "aoss:GetSecurityPolicy"
            ],
            "Resource": "*"
        }
    ]
}
```

Prioritas kebijakan

Mungkin ada situasi di mana aturan kebijakan jaringan tumpang tindih, di dalam atau di seluruh kebijakan. Ketika ini terjadi, aturan yang menentukan akses publik mengesampingkan aturan yang menentukan akses pribadi untuk koleksi apa pun yang umum untuk kedua aturan tersebut.

Misalnya, dalam kebijakan berikut, kedua aturan menetapkan akses jaringan ke `finance` koleksi, tetapi satu aturan menentukan akses VPC sementara yang lain menentukan akses publik. Dalam situasi ini, akses publik mengesampingkan akses VPC hanya untuk pengumpulan keuangan (karena ada di kedua aturan), sehingga pengumpulan keuangan akan dapat diakses dari jaringan publik. Koleksi penjualan akan memiliki akses VPC dari titik akhir yang ditentukan.

```
[
   {
      "Description":"Rule 1",
      "Rules":[
         {
            "ResourceType":"collection",
            "Resource":[
               "collection/sales",
               "collection/finance"
            ]
         }
      ],
      "AllowFromPublic":false,
      "SourceVPCEs":[
         "vpce-050f79086ee71ac05"
      ]
   },
   {
      "Description":"Rule 2",
      "Rules":[
         {
            "ResourceType":"collection",
            "Resource":[
               "collection/finance"
            ]
         }
      ],
      "AllowFromPublic":true
   }
]
```

Jika beberapa titik akhir VPC dari aturan yang berbeda berlaku untuk koleksi, aturan bersifat aditif dan koleksi akan dapat diakses dari semua titik akhir yang ditentukan. Jika Anda menyetel `AllowFromPublic` ke `true` tetapi juga menyediakan satu atau lebih `SourceVPCEs` atau`SourceServices`, OpenSearch Tanpa Server mengabaikan titik akhir VPC dan pengidentifikasi layanan, dan koleksi terkait akan memiliki akses publik.

Membuat kebijakan jaringan (konsol)

Kebijakan jaringan dapat berlaku untuk kebijakan yang ada serta kebijakan future. Kami menyarankan Anda membuat kebijakan jaringan sebelum mulai membuat koleksi.

Untuk membuat kebijakan OpenSearch jaringan Tanpa Server

1. Buka konsol OpenSearch Layanan Amazon di https://console.aws.amazon.com/aos/home.
2. Di panel navigasi kiri, perluas Tanpa Server dan pilih Kebijakan jaringan.
3. Pilih Buat kebijakan jaringan.
4. Berikan nama dan deskripsi untuk kebijakan tersebut.
5. Berikan satu atau lebih aturan. Aturan ini menentukan izin akses untuk koleksi OpenSearch Tanpa Server dan titik akhir Dasbornya OpenSearch .

   Setiap aturan berisi elemen-elemen berikut:

Elemen	Deskripsi
Nama aturan	Nama yang menggambarkan isi aturan. Misalnya, “Akses VPC untuk tim pemasaran”.
Jenis akses	Pilih akses publik atau pribadi. Kemudian, pilih salah satu atau kedua hal berikut: • Titik akhir VPC untuk akses — Tentukan satu atau beberapa titik akhir VPC yang dikelola Tanpa Server — titik akhir OpenSearch VPC yang dikelola. • Layanan AWS akses pribadi — Pilih satu atau lebih yang didukung Layanan AWS.

Elemen	Deskripsi
Jenis sumber daya	Pilih apakah akan menyediakan akses ke OpenSearch titik akhir (yang memungkinkan melakukan panggilan ke OpenSearch API), ke OpenSearch Dasbor (yang memungkin kan akses ke visualisasi dan antarmuka pengguna untuk OpenSearch plugin), atau keduanya. **Note** Layanan AWS akses pribadi hanya berlaku untuk OpenSearch titik akhir koleksi, bukan ke titik akhir OpenSearch Dasbor. Bahkan jika Anda memilih OpenSearch Dasbor, hanya Layanan AWS dapat diberikan akses endpoint.

Untuk setiap jenis sumber daya yang Anda pilih, Anda dapat memilih koleksi yang ada untuk menerapkan pengaturan kebijakan, dan/atau membuat satu atau beberapa pola sumber daya. Pola sumber daya terdiri dari awalan dan wildcard (*), dan menentukan koleksi mana yang akan diterapkan setelan kebijakan.

Misalnya, jika Anda menyertakan pola yang disebut`Marketing*`, koleksi baru atau yang sudah ada yang namanya dimulai dengan “Pemasaran” akan memiliki pengaturan jaringan dalam kebijakan ini secara otomatis diterapkan padanya. Satu wildcard (*) menerapkan kebijakan untuk semua koleksi saat ini dan yang akan datang.

Selain itu, Anda dapat menentukan nama koleksi future tanpa wildcard, sepertI`Finance`. OpenSearch Tanpa server akan menerapkan pengaturan kebijakan ke koleksi yang baru dibuat dengan nama persis itu.

6. Jika Anda puas dengan konfigurasi kebijakan, pilih Buat.

Membuat kebijakan jaringan (AWS CLI)

Untuk membuat kebijakan jaringan menggunakan operasi API OpenSearch Tanpa Server, Anda menentukan aturan dalam format JSON. CreateSecurityPolicyPermintaan menerima kebijakan sebaris dan file.json. Semua koleksi dan pola harus berbentuk`collection/<collection name|pattern>`.

> **Note**
>
> Jenis sumber daya `dashboards` hanya mengizinkan izin ke OpenSearch Dasbor, tetapi agar OpenSearch Dasbor berfungsi, Anda juga harus mengizinkan akses koleksi dari sumber yang sama. Lihat kebijakan kedua di bawah ini untuk contoh.

Untuk menentukan akses pribadi, sertakan salah satu atau kedua elemen berikut:

- `SourceVPCEs`— Tentukan satu atau lebih titik akhir VPC yang OpenSearch dikelola tanpa server.
- `SourceServices`— Tentukan pengenal satu atau lebih yang didukung Layanan AWS. Saat ini, pengidentifikasi layanan berikut didukung:
  - `bedrock.amazonaws.com`— Batuan Dasar Amazon

Contoh kebijakan jaringan berikut menyediakan akses pribadi, ke titik akhir VPC dan Amazon Bedrock, ke titik akhir pengumpulan hanya untuk koleksi yang dimulai dengan awalan. `log*` Pengguna yang diautentikasi tidak dapat masuk ke OpenSearch Dasbor; mereka hanya dapat mengakses titik akhir koleksi secara terprogram.

```
[
   {
      "Description":"Private access for log collections",
      "Rules":[
         {
            "ResourceType":"collection",
            "Resource":[
               "collection/log*"
            ]
         }
      ],
      "AllowFromPublic":false,
      "SourceVPCEs":[
```

```
            "vpce-050f79086ee71ac05"
         ],
         "SourceServices":[
            "bedrock.amazonaws.com"
         ],
      }
]
```

Kebijakan berikut menyediakan akses publik ke OpenSearch titik akhir dan OpenSearch Dasbor untuk satu koleksi bernama. `finance` Jika koleksi tidak ada, pengaturan jaringan akan diterapkan ke koleksi jika dan ketika itu dibuat.

```
[
   {
      "Description":"Public access for finance collection",
      "Rules":[
         {
            "ResourceType":"dashboard",
            "Resource":[
               "collection/finance"
            ]
         },
         {
            "ResourceType":"collection",
            "Resource":[
               "collection/finance"
            ]
         }
      ],
      "AllowFromPublic":true
   }
]
```

Permintaan berikut membuat kebijakan jaringan di atas:

```
aws opensearchserverless create-security-policy \
    --name sales-inventory \
    --type network \
    --policy "[{\"Description\":\"Public access for finance collection\",\"Rules
\":[{\"ResourceType\":\"dashboard\",\"Resource\":[\"collection\/finance\"]},
{\"ResourceType\":\"collection\",\"Resource\":[\"collection\/finance\"]}],
\"AllowFromPublic\":true}]"
```

Untuk menyediakan kebijakan dalam file JSON, gunakan format `--policy file://my-policy.json`

Melihat kebijakan jaringan

Sebelum membuat koleksi, Anda mungkin ingin melihat pratinjau kebijakan jaringan yang ada di akun Anda untuk melihat mana yang memiliki pola sumber daya yang cocok dengan nama koleksi Anda. ListSecurityPoliciesPermintaan berikut mencantumkan semua kebijakan jaringan di akun Anda:

```
aws opensearchserverless list-security-policies --type network
```

Permintaan mengembalikan informasi tentang semua kebijakan jaringan yang dikonfigurasi. Untuk melihat aturan pola yang ditentukan dalam satu kebijakan tertentu, cari informasi kebijakan dalam konten `securityPolicySummaries` elemen dalam respons. Perhatikan `name` dan `type` kebijakan ini dan gunakan properti ini dalam GetSecurityPolicypermintaan untuk menerima tanggapan dengan rincian kebijakan berikut:

```
{
    "securityPolicyDetail": [
        {
            "type": "network",
            "name": "my-policy",
            "policyVersion": "MTY2MzY5MTY1MDA3Ml8x",
            "policy": "[{\"Description\":\"My network policy rule\",\"Rules\":
[{\"ResourceType\":\"dashboard\",\"Resource\":[\"collection/*\"]}],\"AllowFromPublic
\":true}]",
            "createdDate": 1663691650072,
            "lastModifiedDate": 1663691650072
        }
    ]
}
```

Untuk melihat informasi terperinci tentang kebijakan tertentu, gunakan GetSecurityPolicyperintah.

Memperbarui kebijakan jaringan

Saat Anda memodifikasi titik akhir VPC atau penunjukan akses publik untuk jaringan, semua koleksi terkait akan terpengaruh. Untuk memperbarui kebijakan jaringan di konsol OpenSearch Tanpa Server, perluas Kebijakan jaringan, pilih kebijakan yang akan diubah, dan pilih Edit. Buat perubahan dan pilih Simpan.

Untuk memperbarui kebijakan jaringan menggunakan API OpenSearch Tanpa Server, gunakan perintah. UpdateSecurityPolicy Anda harus menyertakan versi kebijakan dalam permintaan. Anda dapat mengambil versi kebijakan dengan menggunakan `GetSecurityPolicy` perintah `ListSecurityPolicies` atau. Menyertakan versi kebijakan terbaru memastikan bahwa Anda tidak secara tidak sengaja mengesampingkan perubahan yang dilakukan oleh orang lain.

Permintaan berikut memperbarui kebijakan jaringan dengan dokumen JSON kebijakan baru:

```
aws opensearchserverless update-security-policy \
    --name sales-inventory \
    --type network \
    --policy-version MTY2MzY5MTY1MDA3Ml8x \
    --policy file://my-new-policy.json
```

Menghapus kebijakan jaringan

Sebelum Anda dapat menghapus kebijakan jaringan, Anda harus melepaskannya dari semua koleksi. Untuk menghapus kebijakan di konsol OpenSearch Tanpa Server, pilih kebijakan dan pilih Hapus.

Anda juga dapat menggunakan DeleteSecurityPolicyperintah:

```
aws opensearchserverless delete-security-policy --name my-policy --type network
```

Kontrol akses data untuk Amazon Tanpa OpenSearch Server

Dengan kontrol akses data di Amazon OpenSearch Tanpa Server, Anda dapat mengizinkan pengguna mengakses koleksi dan indeks, terlepas dari mekanisme akses atau sumber jaringannya. Anda dapat memberikan akses ke peran IAM dan identitas SALL.

Anda mengelola izin akses melalui kebijakan akses data, yang berlaku untuk koleksi dan sumber daya indeks. Kebijakan akses data membantu Anda mengelola koleksi dalam skala besar dengan secara otomatis menetapkan izin akses ke koleksi dan indeks yang cocok dengan pola tertentu. Beberapa kebijakan akses data dapat diterapkan ke satu sumber daya. Perhatikan bahwa Anda harus memiliki kebijakan akses data untuk koleksi Anda untuk mengakses URL OpenSearch Dasbor Anda.

Topik

- Kebijakan akses data versus kebijakan IAM
- Izin IAM yang diperlukan

- Sintaksis kebijakan
- Izin kebijakan yang didukung
- Contoh kumpulan data di Dasbor OpenSearch
- Membuat kebijakan akses data (konsol)
- Membuat kebijakan akses data (AWS CLI)
- Melihat kebijakan akses data
- Memperbarui kebijakan akses data
- Menghapus kebijakan akses data

Kebijakan akses data versus kebijakan IAM

Kebijakan akses data secara logis terpisah dari kebijakan AWS Identity and Access Management (IAM). Izin IAM mengontrol akses ke operasi API tanpa server, seperti dan. `CreateCollection` `ListAccessPolicies` Kebijakan akses data mengontrol akses ke OpenSearch operasi yang didukung OpenSearch Tanpa Server, seperti `PUT <index>` atau. `GET _cat/indices`

Izin IAM yang mengontrol akses ke operasi API kebijakan akses data, seperti `aoss:CreateAccessPolicy` dan `aoss:GetAccessPolicy` (dijelaskan di bagian berikutnya), tidak memengaruhi izin yang ditentukan dalam kebijakan akses data.

Misalnya, kebijakan IAM menyangkal pengguna membuat kebijakan akses data untuk`collection-a`, tetapi memungkinkan mereka membuat kebijakan akses data untuk semua koleksi ()*:

```
{
    "Version": "2012-10-17",
    "Statement": [
        {
            "Effect": "Deny",
            "Action": [
                "aoss:CreateAccessPolicy"
            ],
            "Resource": "*",
            "Condition": {
                "StringLike": {
                    "aoss:collection": "collection-a"
                }
            }
        },
```

```
        {
            "Effect": "Allow",
            "Action": [
                "aoss:CreateAccessPolicy"
            ],
            "Resource": "*"
        }
    ]
}
```

Jika pengguna membuat kebijakan akses data yang mengizinkan izin tertentu untuk semua koleksi (`collection/*`atau`index/*/*`) kebijakan tersebut akan berlaku untuk semua koleksi, termasuk koleksi A.

> **Important**
>
> Pemberian izin dalam kebijakan akses data tidak cukup untuk mengakses data dalam koleksi Tanpa OpenSearch Server Anda. Prinsipal terkait juga harus diberikan akses ke izin `aoss:APIAccessAll` IAM dan. `aoss:DashboardAccessAll` Kedua izin memberikan akses penuh ke sumber daya koleksi, sementara izin Dasbor juga menyediakan akses ke OpenSearch Dasbor. Jika prinsipal tidak memiliki kedua izin IAM ini, mereka akan menerima 403 kesalahan saat mencoba mengirim permintaan ke koleksi. Untuk informasi selengkapnya, lihat the section called “Menggunakan operasi OpenSearch API”.

Izin IAM yang diperlukan

Kontrol akses data untuk OpenSearch Tanpa Server menggunakan izin IAM berikut. Anda dapat menentukan kondisi IAM untuk membatasi pengguna ke nama kebijakan akses tertentu.

- `aoss:CreateAccessPolicy`— Buat kebijakan akses.
- `aoss:ListAccessPolicies`— Daftar semua kebijakan akses.
- `aoss:GetAccessPolicy`— Lihat detail tentang kebijakan akses tertentu.
- `aoss:UpdateAccessPolicy`— Memodifikasi kebijakan akses.
- `aoss:DeleteAccessPolicy`— Hapus kebijakan akses.

Kebijakan akses berbasis identitas berikut memungkinkan pengguna untuk melihat semua kebijakan akses, dan memperbarui kebijakan yang berisi pola sumber daya. `collection/logs`

```
{
    "Version": "2012-10-17",
    "Statement": [
        {
            "Action": [
                "aoss:ListAccessPolicies",
                "aoss:GetAccessPolicy"
            ],
            "Effect": "Allow",
            "Resource": "*"
        },
        {
            "Action": [
                "aoss:UpdateAccessPolicy"
            ],
            "Effect": "Allow",
            "Resource": "*",
            "Condition": {
                "StringEquals": {
                    "aoss:collection": [
                        "logs"
                    ]
                }
            }
        }
    ]
}
```

Sintaksis kebijakan

Kebijakan akses data mencakup seperangkat aturan, masing-masing dengan elemen berikut:

Elemen	Deskripsi
`ResourceType`	Jenis sumber daya (koleksi atau indeks) yang diterapkan izin. Izin alias dan template berada di tingkat pengumpulan, sementara izin untuk membuat, memodifikasi, dan mencari data berada pada tingkat indeks. Untuk informasi selengkapnya, lihat Izin kebijakan yang didukung.

Elemen	Deskripsi
`Resource`	Daftar nama dan/atau pola sumber daya. Pola adalah awalan yang diikuti oleh wildcard (*), yang memungkinkan izin terkait diterapkan ke beberapa sumber daya. • Koleksi mengambil format`collection/` *`<name|pattern>`* . • Indeks mengambil format`index/`*`<collection-name|pattern>`* `/`*`<index-name|pattern/>`* .
`Permission`	Daftar izin untuk diberikan untuk sumber daya yang ditentukan. Untuk daftar lengkap izin dan operasi API yang diizinkan, lihatthe section called “Operasi dan izin OpenSearch API yang didukung”.
`Principal`	Daftar satu atau lebih kepala sekolah untuk memberikan akses ke. Prinsipal dapat berupa ARN peran IAM atau identitas SALL. Prinsipal ini harus berada dalam arus. Akun AWS Akses lintas akun tidak didukung.

Kebijakan contoh berikut memberikan izin alias dan templat ke koleksi yang dipanggil`autopartsinventory`, serta koleksi apa pun yang dimulai dengan awalan. `sales*` Ini juga memberikan izin baca dan tulis ke semua indeks dalam `autopartsinventory` koleksi, dan indeks apa pun dalam `salesorders` koleksi yang dimulai dengan awalan. `orders*`

```
[
   {
      "Description": "Rule 1",
      "Rules":[
         {
            "ResourceType":"collection",
            "Resource":[
               "collection/autopartsinventory",
               "collection/sales*"
            ],
            "Permission":[
               "aoss:CreateCollectionItems",
               "aoss:UpdateCollectionItems",
               "aoss:DescribeCollectionItems"
            ]
         },
```

```
         {
            "ResourceType":"index",
            "Resource":[
               "index/autopartsinventory/*",
               "index/salesorders/orders*"
            ],
            "Permission":[
               "aoss:*"
            ]
         }
      ],
      "Principal":[
         "arn:aws:iam::123456789012:user/Dale",
         "arn:aws:iam::123456789012:role/RegulatoryCompliance",
         "saml/123456789012/myprovider/user/Annie",
         "saml/123456789012/anotherprovider/group/Accounting"
      ]
   }
]
```

Anda tidak dapat secara eksplisit menolak akses dalam kebijakan. Oleh karena itu, semua izin kebijakan bersifat aditif. Misalnya, jika satu kebijakan memberikan penggunaaoss:ReadDocument, dan kebijakan lain memberikanaoss:WriteDocument, pengguna akan memiliki kedua izin tersebut. Jika kebijakan ketiga memberikan pengguna yang samaaoss:*, maka pengguna dapat melakukan semua tindakan pada indeks terkait; izin yang lebih ketat tidak akan mengesampingkan yang kurang membatasi.

Izin kebijakan yang didukung

Izin berikut didukung dalam kebijakan akses data. Untuk operasi OpenSearch API yang diizinkan oleh setiap izin, lihatthe section called “Operasi dan izin OpenSearch API yang didukung”.

Izin koleksi

- aoss:CreateCollectionItems
- aoss:DeleteCollectionItems
- aoss:UpdateCollectionItems
- aoss:DescribeCollectionItems
- aoss:*

Izin indeks

- `aoss:ReadDocument`
- `aoss:WriteDocument`
- `aoss:CreateIndex`
- `aoss:DeleteIndex`
- `aoss:UpdateIndex`
- `aoss:DescribeIndex`
- `aoss:*`

Contoh kumpulan data di Dasbor OpenSearch

OpenSearch Dasbor menyediakan contoh kumpulan data yang dilengkapi dengan visualisasi, dasbor, dan alat lain untuk membantu Anda menjelajahi Dasbor sebelum menambahkan data Anda sendiri. Untuk membuat indeks dari data sampel ini, Anda memerlukan kebijakan akses data yang menyediakan izin ke kumpulan data yang ingin Anda gunakan. Kebijakan berikut menggunakan wildcard (*) untuk memberikan izin ke ketiga kumpulan data sampel.

```
[
  {
    "Rules": [
      {
        "Resource": [
          "index/<collection-name>/opensearch_dashboards_sample_data_*"
        ],
        "Permission": [
          "aoss:CreateIndex",
          "aoss:DescribeIndex",
          "aoss:ReadDocument"
        ],
        "ResourceType": "index"
      }
    ],
    "Principal": [
      "arn:aws:iam::<account-id>:user/<user>"
    ]
  }
]
```

Membuat kebijakan akses data (konsol)

Anda dapat membuat kebijakan akses data menggunakan editor visual, atau dalam format JSON. Setiap koleksi baru yang cocok dengan salah satu pola yang ditentukan dalam kebijakan akan diberikan izin terkait saat Anda membuat koleksi.

Untuk membuat kebijakan OpenSearch akses data Tanpa Server

1. Buka konsol OpenSearch Layanan Amazon di https://console.aws.amazon.com/aos/home.
2. Di panel navigasi kiri, perluas Tanpa Server dan pilih Kontrol akses data.
3. Pilih Buat kebijakan akses.
4. Berikan nama dan deskripsi untuk kebijakan tersebut.
5. Berikan nama untuk aturan pertama dalam kebijakan Anda. Misalnya, “Akses koleksi log”.
6. Pilih Tambahkan prinsipal dan pilih satu atau beberapa peran IAM atau pengguna dan grup SAFL untuk menyediakan akses data.

   > **Note**
   >
   > Untuk memilih prinsipal dari menu tarik-turun, Anda harus memiliki dan izin (untuk kepala sekolah IAM) `iam:ListUsers` dan `iam:ListRoles` izin (untuk identitas SAFL). `aoss:ListSecurityConfigs`

7. Pilih Grant dan pilih alias, template, dan izin indeks untuk memberikan prinsipal terkait. Untuk daftar lengkap izin dan akses yang mereka izinkan, lihatthe section called “Operasi dan izin OpenSearch API yang didukung”.
8. (Opsional) Konfigurasikan aturan tambahan untuk kebijakan tersebut.
9. Pilih Buat. Mungkin ada sekitar satu menit jeda waktu antara saat Anda membuat kebijakan dan saat izin diberlakukan. Jika dibutuhkan lebih dari 5 menit, hubungi AWS Support.

> **Important**
>
> Jika kebijakan Anda hanya menyertakan izin indeks (dan tidak ada izin pengumpulan), Anda mungkin masih melihat pesan untuk pencocokan koleksi yang menyatakan. `Collection cannot be accessed yet. Configure data access policies so that users can access the data within this collection` Anda dapat mengabaikan

peringatan ini. Prinsipal yang diizinkan masih dapat melakukan operasi terkait indeks yang ditetapkan pada koleksi.

Membuat kebijakan akses data (AWS CLI)

Untuk membuat kebijakan akses data menggunakan API OpenSearch Tanpa Server, gunakan perintah. `CreateAccessPolicy` Perintah menerima kebijakan inline dan file.json. Kebijakan sebaris harus dikodekan sebagai string lolos JSON.

Permintaan berikut membuat kebijakan akses data:

```
aws opensearchserverless create-access-policy \
    --name marketing \
    --type data \
    --policy "[{\"Rules\":[{\"ResourceType\":\"collection\",\"Resource\":
[\"collection/autopartsinventory\",\"collection/sales*\"],\"Permission\":
[\"aoss:UpdateCollectionItems\"]},{\"ResourceType\":\"index\",\"Resource\":
[\"index/autopartsinventory/*\",\"index/salesorders/orders*\"],\"Permission
\":[\"aoss:ReadDocument\",\"aoss:DescribeIndex\"]}],\"Principal\":
[\"arn:aws:iam::123456789012:user/Shaheen\"]}]"
```

Untuk menyediakan kebijakan dalam file.json, gunakan formatnya. `--policy file://my-policy.json`

Prinsipal yang termasuk dalam kebijakan sekarang dapat menggunakan OpenSearch operasi yang diberikan akses kepada mereka.

Melihat kebijakan akses data

Sebelum membuat koleksi, Anda mungkin ingin melihat pratinjau kebijakan akses data yang ada di akun Anda untuk melihat mana yang memiliki pola sumber daya yang cocok dengan nama koleksi Anda. ListAccessPoliciesPermintaan berikut mencantumkan semua kebijakan akses data di akun Anda:

```
aws opensearchserverless list-access-policies --type data
```

Permintaan mengembalikan informasi tentang semua kebijakan akses data yang dikonfigurasi. Untuk melihat aturan pola yang ditentukan dalam satu kebijakan tertentu, cari informasi kebijakan dalam konten `accessPolicySummaries` elemen dalam respons. Perhatikan `name` dan `type` kebijakan

ini dan gunakan properti ini dalam GetAccessPolicypermintaan untuk menerima tanggapan dengan rincian kebijakan berikut:

```
{
    "accessPolicyDetails": [
        {
            "type": "data",
            "name": "my-policy",
            "policyVersion": "MTY2NDA1NDE4MDg1OF8x",
            "description": "My policy",
            "policy": "[{\"Rules\":[{\"ResourceType\":\"collection\",
\"Resource\":[\"collection/autopartsinventory\",\"collection/sales*\"],
\"Permission\":[\"aoss:UpdateCollectionItems\"]},{\"ResourceType\":\"index\",
\"Resource\":[\"index/autopartsinventory/*\",\"index/salesorders/orders*\"],
\"Permission\":[\"aoss:ReadDocument\",\"aoss:DescribeIndex\"]}],\"Principal\":
[\"arn:aws:iam::123456789012:user/Shaheen\"]}]",
            "createdDate": 1664054180858,
            "lastModifiedDate": 1664054180858
        }
    ]
}
```

Anda dapat menyertakan filter sumber daya untuk membatasi hasil pada kebijakan yang berisi koleksi atau indeks tertentu:

```
aws opensearchserverless list-access-policies --type data --resource
 "index/autopartsinventory/*"
```

Untuk melihat detail tentang kebijakan tertentu, gunakan GetAccessPolicyperintah.

Memperbarui kebijakan akses data

Saat Anda memperbarui kebijakan akses data, semua koleksi terkait akan terpengaruh. Untuk memperbarui kebijakan akses data di konsol OpenSearch Tanpa Server, pilih Kontrol akses data, pilih kebijakan yang akan diubah, dan pilih Edit. Buat perubahan dan pilih Simpan.

Untuk memperbarui kebijakan akses data menggunakan API OpenSearch Tanpa Server, kirim permintaan. `UpdateAccessPolicy` Anda harus menyertakan versi kebijakan, yang dapat Anda ambil menggunakan `GetAccessPolicy` perintah `ListAccessPolicies` atau. Menyertakan versi kebijakan terbaru memastikan bahwa Anda tidak secara tidak sengaja mengesampingkan perubahan yang dilakukan oleh orang lain.

UpdateAccessPolicyPermintaan berikut memperbarui kebijakan akses data dengan dokumen JSON kebijakan baru:

```
aws opensearchserverless update-access-policy \
    --name sales-inventory \
    --type data \
    --policy-version MTY2NDA1NDE4MDg1OF8x \
    --policy file://my-new-policy.json
```

Mungkin ada jeda waktu beberapa menit antara saat Anda memperbarui kebijakan dan saat izin baru diberlakukan.

Menghapus kebijakan akses data

Saat Anda menghapus kebijakan akses data, semua koleksi terkait kehilangan akses yang ditentukan dalam kebijakan. Pastikan bahwa pengguna IAM dan SALL Anda memiliki akses yang sesuai ke koleksi sebelum Anda menghapus kebijakan. Untuk menghapus kebijakan di konsol OpenSearch Tanpa Server, pilih kebijakan dan pilih Hapus.

Anda juga dapat menggunakan DeleteAccessPolicyperintah:

```
aws opensearchserverless delete-access-policy --name my-policy --type data
```

Akses Amazon OpenSearch Tanpa Server menggunakan titik akhir antarmuka () AWS PrivateLink

Anda dapat menggunakan AWS PrivateLink untuk membuat koneksi pribadi antara VPC Anda dan Amazon OpenSearch Tanpa Server. Anda dapat mengakses OpenSearch Tanpa Server seolah-olah berada di VPC Anda, tanpa menggunakan gateway internet, perangkat NAT, koneksi VPN, atau koneksi. AWS Direct Connect Instans di VPC Anda tidak memerlukan alamat IP publik untuk OpenSearch mengakses Tanpa Server.

Anda membuat koneksi pribadi ini dengan membuat titik akhir antarmuka, didukung olehAWS PrivateLink. Kami membuat antarmuka jaringan endpoint di setiap subnet yang Anda tentukan untuk titik akhir antarmuka. Ini adalah antarmuka jaringan yang dikelola pemohon yang berfungsi sebagai titik masuk untuk lalu lintas yang ditujukan untuk Tanpa Server. OpenSearch

Untuk informasi selengkapnya, lihat Mengakses Layanan AWS melalui AWS PrivateLink di Panduan AWS PrivateLink.

Topik

Resolusi DNS dari titik akhir koleksi

Saat Anda membuat titik akhir VPC, layanan akan membuat zona host Amazon Route 53 pribadi baru dan menempelkannya ke VPC. Zona host pribadi ini terdiri dari catatan untuk menyelesaikan catatan DNS wildcard untuk koleksi OpenSearch Tanpa Server (`*.aoss.us-east-1.amazonaws.com`) ke alamat antarmuka yang digunakan untuk titik akhir. Anda hanya memerlukan satu titik akhir OpenSearch VPC Tanpa Server di VPC untuk mengakses setiap dan semua koleksi dan Dasbor di masing-masing. Wilayah AWS Setiap VPC dengan titik akhir untuk OpenSearch Tanpa Server memiliki zona host pribadinya sendiri yang terpasang.

OpenSearch Tanpa server juga membuat catatan DNS wildcard Route 53 publik untuk semua koleksi di Wilayah. Nama DNS diselesaikan ke alamat IP publik Tanpa OpenSearch Server. Klien di VPC yang tidak memiliki titik akhir VPC OpenSearch Tanpa Server atau klien di jaringan publik dapat menggunakan resolver Route 53 publik dan mengakses koleksi dan Dasbor dengan alamat IP tersebut.

Alamat penyelesai DNS untuk VPC tertentu adalah alamat IP kedua dari VPC CIDR. Setiap klien di VPC perlu menggunakan resolver itu untuk mendapatkan alamat titik akhir VPC untuk koleksi apa pun. Penyelesai menggunakan zona host pribadi yang dibuat oleh OpenSearch Tanpa Server. Cukup menggunakan resolver itu untuk semua koleksi di akun apa pun. Dimungkinkan juga untuk menggunakan resolver VPC untuk beberapa titik akhir koleksi dan resolver publik untuk yang lain, meskipun biasanya tidak diperlukan.

VPC dan kebijakan akses jaringan

Untuk memberikan izin jaringan ke OpenSearch API dan Dasbor untuk koleksi Anda, Anda dapat menggunakan kebijakan akses jaringan OpenSearch Tanpa Server. Anda dapat mengontrol akses

jaringan ini baik dari titik akhir VPC Anda atau internet publik. Karena kebijakan jaringan Anda hanya mengontrol izin lalu lintas, Anda juga harus menyiapkan kebijakan akses data yang menentukan izin untuk beroperasi pada data dalam koleksi dan indeksnya. Pikirkan titik akhir OpenSearch VPC Tanpa Server sebagai titik akses ke layanan, kebijakan akses jaringan sebagai titik akses tingkat jaringan ke koleksi dan Dasbor, dan kebijakan akses data sebagai titik akses untuk kontrol akses berbutir halus untuk operasi apa pun pada data dalam pengumpulan.

Karena Anda dapat menentukan beberapa ID titik akhir VPC dalam kebijakan jaringan, sebaiknya Anda membuat titik akhir VPC untuk setiap VPC yang perlu mengakses koleksi. VPC ini dapat dimiliki oleh AWS akun yang berbeda dari akun yang memiliki koleksi OpenSearch Tanpa Server dan kebijakan jaringan. Kami tidak menyarankan Anda membuat peering VPC-ke-VPC atau solusi proxy lainnya antara dua akun sehingga VPC satu akun dapat menggunakan titik akhir VPC akun lain. Ini kurang aman dan hemat biaya dibandingkan setiap VPC yang memiliki endpoint sendiri. VPC pertama tidak akan mudah terlihat oleh admin VPC lain, yang telah mengatur akses ke titik akhir VPC itu dalam kebijakan jaringan.

VPC dan kebijakan endpoint

Amazon OpenSearch Serverless mendukung kebijakan endpoint untuk VPC. Kebijakan endpoint adalah kebijakan berbasis sumber daya IAM yang Anda lampirkan ke titik akhir VPC untuk mengontrol AWS prinsipal mana yang dapat menggunakan titik akhir untuk mengakses layanan Anda. AWS Untuk informasi selengkapnya, lihat Mengontrol akses ke titik akhir VPC menggunakan kebijakan titik akhir.

Untuk menggunakan kebijakan endpoint, Anda harus terlebih dahulu membuat endpoint antarmuka. Anda dapat membuat titik akhir antarmuka menggunakan konsol OpenSearch Tanpa Server atau API Tanpa Server. OpenSearch Setelah membuat titik akhir antarmuka, Anda perlu menambahkan kebijakan titik akhir ke titik akhir. Untuk informasi selengkapnya, lihat Mengakses Amazon OpenSearch Tanpa Server menggunakan titik akhir antarmuka (). AWS PrivateLink

> **Note**
>
> Anda tidak dapat menentukan kebijakan titik akhir secara langsung di konsol OpenSearch Layanan.

Kebijakan endpoint tidak mengesampingkan atau mengganti kebijakan berbasis identitas lainnya, kebijakan berbasis sumber daya, kebijakan jaringan, atau kebijakan akses data yang mungkin telah

Anda konfigurasikan. Untuk informasi selengkapnya tentang memperbarui kebijakan titik akhir, lihat Mengontrol akses ke titik akhir VPC menggunakan kebijakan titik akhir.

Secara default, kebijakan endpoint memberikan akses penuh ke titik akhir VPC Anda.

```
{
    "Statement": [
        {
            "Effect": "Allow",
            "Principal": "*",
            "Action": "*",
            "Resource": "*"
        }
    ]
}
```

Meskipun kebijakan titik akhir VPC default memberikan akses titik akhir penuh, Anda dapat mengonfigurasi kebijakan titik akhir VPC untuk mengizinkan akses ke peran dan pengguna tertentu. Untuk melakukan ini, lihat contoh berikut:

```
{
    "Version": "2012-10-17",
    "Statement": [
        {
            "Effect": "Allow",
            "Principal": {
                "AWS": [
                    "123456789012",
                    "987654321098"
                ]
            },
            "Action": "*",
            "Resource": "*"
        }
    ]
}
```

Anda dapat menentukan koleksi OpenSearch Tanpa Server yang akan disertakan sebagai elemen bersyarat dalam kebijakan titik akhir VPC Anda. Untuk melakukan ini, lihat contoh berikut:

```
{
```

```
    "Version": "2012-10-17",
    "Statement": [
        {
            "Effect": "Allow",
            "Principal": "*",
            "Action": "*",
            "Resource": "*",
            "Condition": {
                "StringEquals": {
                    "aws:CollectionName": [
                        "coll-abc"
                    ]
                }
            }
        }
    ]
}
```

Anda dapat menggunakan identitas SAFL dalam kebijakan titik akhir VPC Anda untuk menentukan akses titik akhir VPC. Anda harus menggunakan wildcard (*) di bagian utama kebijakan titik akhir VPC Anda. Untuk melakukan ini, lihat contoh berikut:

```
{
    "Version": "2012-10-17",
    "Statement": [
        {
            "Effect": "Allow",
            "Principal": "*",
            "Action": "*",
            "Resource": "*",
            "Condition": {
                "ForAnyValue:StringEquals": {
                    "aws:SamlGroups": [
                        "saml/123456789012/idp123/group/football",
                        "saml/123456789012/idp123/group/soccer",
                        "saml/123456789012/idp123/group/cricket"
                    ]
                }
            }
        }
    ]
}
```

Selain itu, Anda dapat mengonfigurasi kebijakan titik akhir Anda untuk menyertakan kebijakan utama SALL tertentu. Untuk melakukan ini, lihat yang berikut ini:

```
{
    "Version": "2012-10-17",
    "Statement": [
        {
            "Effect": "Allow",
            "Principal": "*",
            "Action": "*",
            "Resource": "*",
            "Condition": {
                "StringEquals": {
                    "aws:SamlPrincipal": [
                        "saml/123456789012/idp123/user/user1234"]
                    }
                }
            }
        ]
    }
```

Untuk informasi selengkapnya tentang menggunakan otentikasi SAFL dengan Amazon OpenSearch Tanpa Server, lihat otentikasi SAFL untuk Amazon Tanpa Server. OpenSearch

Anda juga dapat menyertakan pengguna IAM dan SAFL dalam kebijakan titik akhir VPC yang sama. Untuk melakukan ini, lihat contoh berikut:

```
{
    "Version": "2012-10-17",
    "Statement": [
        {
            "Effect": "Allow",
            "Principal": "*",
            "Action": "*",
            "Resource": "*",
            "Condition": {
                "ForAnyValue:StringEquals": {
                    "aws:SamlGroups": [
                        "saml/123456789012/idp123/group/football",
                        "saml/123456789012/idp123/group/soccer",
                        "saml/123456789012/idp123/group/cricket"
                    ]
```

```
                }
            }
        },
        {
            "Effect": "Allow",
            "Principal": {
                "AWS": [
                    "123456789012"
                ]
            },
            "Action": "*",
            "Resource": "*"
        }
    ]
}
```

Pertimbangan

Sebelum Anda menyiapkan titik akhir antarmuka untuk OpenSearch Tanpa Server, pertimbangkan hal berikut:

- OpenSearch Tanpa server mendukung panggilan ke semua operasi OpenSearch API yang didukung (bukan operasi API konfigurasi) melalui titik akhir antarmuka.
- Setelah Anda membuat titik akhir antarmuka untuk OpenSearch Tanpa Server, Anda masih perlu memasukkannya ke dalam kebijakan akses jaringan agar dapat mengakses koleksi tanpa server.
- Secara default, akses penuh ke OpenSearch Tanpa Server diizinkan melalui titik akhir antarmuka. Anda dapat mengaitkan grup keamanan dengan antarmuka jaringan titik akhir untuk mengontrol lalu lintas ke OpenSearch Tanpa Server melalui titik akhir antarmuka.
- Satu Akun AWS dapat memiliki maksimal 50 titik akhir OpenSearch VPC Tanpa Server.
- Jika Anda mengaktifkan akses internet publik ke API atau Dasbor koleksi Anda dalam kebijakan jaringan, koleksi Anda dapat diakses oleh VPC apa pun dan oleh internet publik.
- Jika Anda berada di lokasi dan di luar VPC, Anda tidak dapat menggunakan resolver DNS untuk resolusi titik akhir VPC Tanpa Server OpenSearch secara langsung. Jika Anda memerlukan akses VPN, VPC memerlukan resolver proxy DNS untuk digunakan klien eksternal. Route 53 menyediakan opsi titik akhir masuk yang dapat Anda gunakan untuk menyelesaikan kueri DNS ke VPC dari jaringan lokal atau VPC lain.
- Untuk pertimbangan lain, lihat Pertimbangan dalam Panduan. AWS PrivateLink

Izin diperlukan

Akses VPC untuk OpenSearch Tanpa Server menggunakan izin AWS Identity and Access Management (IAM) berikut. Anda dapat menentukan kondisi IAM untuk membatasi pengguna ke koleksi tertentu.

- `aoss:CreateVpcEndpoint`— Buat titik akhir VPC.
- `aoss:ListVpcEndpoints`— Daftar semua titik akhir VPC.
- `aoss:BatchGetVpcEndpoint`— Lihat detail tentang subset titik akhir VPC.
- `aoss:UpdateVpcEndpoint`— Memodifikasi titik akhir VPC.
- `aoss:DeleteVpcEndpoint`— Hapus titik akhir VPC.

Selain itu, Anda memerlukan izin Amazon EC2 dan Route 53 berikut untuk membuat titik akhir VPC.

- `ec2:CreateTags`
- `ec2:CreateVpcEndpoint`
- `ec2:DeleteVpcEndPoints`
- `ec2:DescribeSecurityGroups`
- `ec2:DescribeSubnets`
- `ec2:DescribeVpcEndpoints`
- `ec2:DescribeVpcs`
- `ec2:ModifyVpcEndPoint`
- `route53:AssociateVPCWithHostedZone`
- `route53:ChangeResourceRecordSets`
- `route53:CreateHostedZone`
- `route53:DeleteHostedZone`
- `route53:GetChange`
- `route53:GetHostedZone`
- `route53:ListHostedZonesByName`
- `route53:ListHostedZonesByVPC`
- `route53:ListResourceRecordSets`

Buat titik akhir antarmuka untuk Tanpa Server OpenSearch

Anda dapat membuat titik akhir antarmuka untuk OpenSearch Tanpa Server menggunakan konsol atau API Tanpa Server. OpenSearch

Untuk membuat titik akhir antarmuka untuk koleksi Tanpa OpenSearch Server

1. Buka konsol OpenSearch Layanan Amazon di https://console.aws.amazon.com/aos/home.
2. Di panel navigasi kiri, perluas Tanpa Server dan pilih titik akhir VPC.
3. Pilih Buat titik akhir VPC.
4. Berikan nama untuk titik akhir.
5. Untuk VPC, pilih VPC tempat Anda akan mengakses Tanpa Server. OpenSearch
6. Untuk Subnet, pilih satu subnet yang akan Anda akses tanpa OpenSearch server.
7. Untuk grup Keamanan, pilih grup keamanan untuk dikaitkan dengan antarmuka jaringan titik akhir. Ini adalah langkah penting di mana Anda membatasi port, protokol, dan sumber untuk lalu lintas masuk yang Anda otorisasi ke titik akhir Anda. Pastikan bahwa aturan grup keamanan memungkinkan sumber daya yang akan menggunakan titik akhir VPC untuk berkomunikasi dengan OpenSearch Tanpa Server untuk berkomunikasi dengan antarmuka jaringan titik akhir.
8. Pilih Buat titik akhir.

Untuk membuat titik akhir VPC menggunakan API OpenSearch Tanpa Server, gunakan perintah. `CreateVpcEndpoint`

> **Note**
>
> Setelah Anda membuat titik akhir, catat ID-nya (misalnya,`vpce-050f79086ee71ac05`. Untuk memberikan akses titik akhir ke koleksi Anda, Anda harus menyertakan ID ini dalam satu atau beberapa kebijakan akses jaringan.

Langkah selanjutnya: Berikan akses endpoint ke koleksi

Setelah membuat titik akhir antarmuka, Anda harus menyediakannya akses ke koleksi melalui kebijakan akses jaringan. Lihat informasi yang lebih lengkap di the section called “Akses jaringan”.

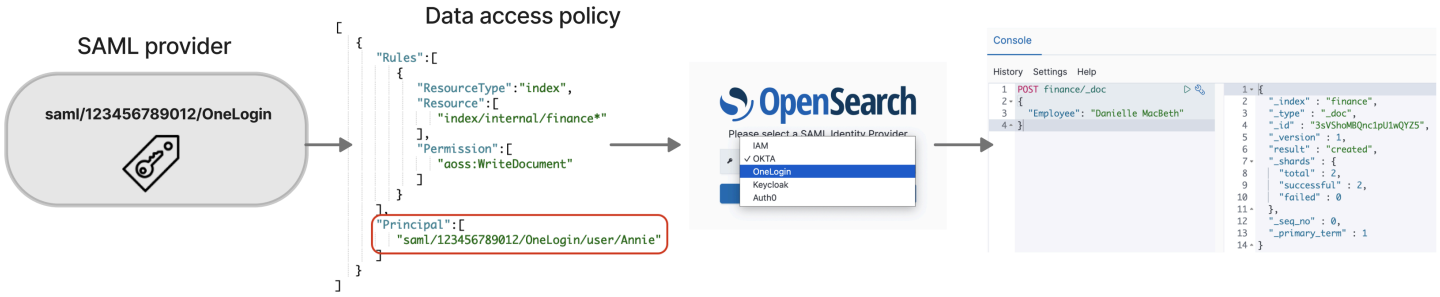
Otentikasi SAMP untuk Amazon Tanpa Server OpenSearch

Dengan autentikasi SAMP untuk Amazon OpenSearch Tanpa Server, Anda dapat menggunakan penyedia identitas yang ada untuk menawarkan sistem masuk tunggal (SSO) untuk titik akhir Dasbor koleksi tanpa server. OpenSearch

Otentikasi SAMP memungkinkan Anda menggunakan penyedia identitas pihak ketiga untuk masuk ke OpenSearch Dasbor untuk mengindeks dan mencari data. OpenSearch Serverless mendukung penyedia yang menggunakan standar SAMP 2.0, seperti IAM Identity Center, Okta, Keycloak, Active Directory Federation Services (AD FS), dan Auth0. Anda dapat mengonfigurasi Pusat Identitas IAM untuk menyinkronkan pengguna dan grup dari sumber identitas lain seperti Okta,, OneLogin dan Microsoft Entra ID. Untuk daftar sumber identitas yang didukung oleh IAM Identity Center dan langkah-langkah untuk mengonfigurasinya, lihat Memulai tutorial di Panduan Pengguna Pusat Identitas IAM.

> **Note**
>
> Otentikasi SAMP hanya untuk mengakses OpenSearch Dasbor melalui browser web. Pengguna yang diautentikasi hanya dapat membuat permintaan ke operasi OpenSearch API melalui Alat Pengembang di OpenSearch Dasbor. Kredensyal SAMP Anda tidak memungkinkan Anda membuat permintaan HTTP langsung ke operasi API. OpenSearch

Untuk mengatur otentikasi SAMP, pertama-tama Anda mengonfigurasi penyedia identitas SAMP (iDP). Anda kemudian menyertakan satu atau beberapa pengguna dari IDP tersebut dalam kebijakan akses data. Kebijakan ini memberikan izin tertentu untuk koleksi dan/atau indeks. Pengguna kemudian dapat masuk ke OpenSearch Dasbor dan melakukan tindakan yang diizinkan dalam kebijakan akses data.



Topik

- Pertimbangan

Pertimbangan

Pertimbangkan hal berikut saat mengonfigurasi otentikasi SAMP:

- Permintaan yang ditandatangani dan dienkripsi tidak didukung.
- Pernyataan terenkripsi tidak didukung.
- Autentikasi dan sign-out yang diprakarsai IDP tidak didukung.

Izin diperlukan

Otentikasi SAMP untuk OpenSearch Tanpa Server menggunakan izin AWS Identity and Access Management (IAM) berikut:

- `aoss:CreateSecurityConfig`— Buat penyedia SAMP.
- `aoss:ListSecurityConfig`— Daftar semua penyedia SAMP di akun saat ini.
- `aoss:GetSecurityConfig`— Lihat informasi penyedia SAMP.
- `aoss:UpdateSecurityConfig`— Memodifikasi konfigurasi penyedia SAMP yang diberikan, termasuk metadata XML.
- `aoss:DeleteSecurityConfig`— Hapus penyedia SAMP.

Kebijakan akses berbasis identitas berikut memungkinkan pengguna untuk mengelola semua konfigurasi IDP:

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

```
    "Statement": [
        {
            "Action": [
                "aoss:CreateSecurityConfig",
                "aoss:DeleteSecurityConfig",
                "aoss:GetSecurityConfig",
                "aoss:UpdateSecurityConfig",
                "aoss:ListSecurityConfigs"
            ],
            "Effect": "Allow",
            "Resource": "*"
        }
    ]
}
```

Perhatikan bahwa `Resource` elemen harus menjadi wildcard.

Membuat penyedia SAMP (konsol)

Langkah-langkah ini menjelaskan cara membuat penyedia SAMP. Ini memungkinkan otentikasi SAMP dengan otentikasi yang dimulai oleh penyedia layanan (SP) untuk Dasbor. OpenSearch Otentikasi yang diprakarsai IDP tidak didukung.

Untuk mengaktifkan otentikasi SAMP untuk Dasbor OpenSearch

1. Masuk ke konsol OpenSearch Layanan Amazon di https://console.aws.amazon.com/aos/home.
2. Di panel navigasi kiri, perluas Tanpa Server dan pilih otentikasi SAMP.
3. Pilih Tambahkan penyedia SAMP.
4. Berikan nama dan deskripsi untuk penyedia.

   > **Note**
   >
   > Nama yang Anda tentukan dapat diakses publik dan akan muncul di menu tarik-turun saat pengguna masuk ke Dasbor. OpenSearch Pastikan nama tersebut mudah dikenali dan tidak mengungkapkan informasi sensitif tentang penyedia identitas Anda.

5. Di bawah Konfigurasi iDP Anda, salin URL assertion consumer service (ACS).
6. Gunakan URL ACS yang baru saja Anda salin untuk mengonfigurasi penyedia identitas Anda. Terminologi dan langkah-langkah bervariasi menurut penyedia. Baca dokumentasi dari penyedia Anda.

Di Okta, misalnya, Anda membuat “aplikasi web SAMP 2.0” dan menentukan URL ACS sebagai URL Single Sign On, URL Penerima, dan URL Tujuan. Untuk Auth0, Anda menentukannya di URL Callback yang Diizinkan.

7. Berikan batasan audiens jika IDP Anda memiliki bidang untuk itu. Pembatasan audiens adalah nilai dalam pernyataan SAMP yang menentukan untuk siapa pernyataan itu dimaksudkan. Untuk OpenSearch Tanpa Server, tentukan. `aws:opensearch:<aws account id>` Misalnya, `aws:opensearch:123456789012`.

   Nama bidang pembatasan audiens bervariasi menurut penyedia. Untuk Okta itu Audience URI (SP Entity ID). Untuk Pusat Identitas IAM itu adalah audiens Aplikasi SAMP.

8. Jika Anda menggunakan IAM Identity Center, Anda juga perlu menentukan pemetaan atribut berikut:`Subject=${user:name}`, dengan format. `unspecified`

9. Setelah Anda konfigurasi, penyedia identitas akan menghasilkan file metadata IdP. File XHTML ini berisi informasi tentang penyedia, seperti sertifikat TLS, titik akhir masuk tunggal, dan ID entitas penyedia identitas.

   Salin teks dalam file metadata iDP dan tempel di bawah Menyediakan metadata dari bidang IDP Anda. Sebagai alternatif, pilih Impor dari file XHTML dan unggah file. File metadata harus terlihat seperti ini:

```
<?xml version="1.0" encoding="UTF-8"?>
<md:EntityDescriptor entityID="entity-id"
 xmlns:md="urn:oasis:names:tc:SAML:2.0:metadata">
  <md:IDPSSODescriptor WantAuthnRequestsSigned="false"
 protocolSupportEnumeration="urn:oasis:names:tc:SAML:2.0:protocol">
    <md:KeyDescriptor use="signing">
      <ds:KeyInfo xmlns:ds="http://www.w3.org/2000/09/xmldsig#">
        <ds:X509Data>
          <ds:X509Certificate>tls-certificate</ds:X509Certificate>
        </ds:X509Data>
      </ds:KeyInfo>s
    </md:KeyDescriptor>
    <md:NameIDFormat>urn:oasis:names:tc:SAML:1.1:nameid-format:unspecified</
md:NameIDFormat>
    <md:NameIDFormat>urn:oasis:names:tc:SAML:1.1:nameid-format:emailAddress</
md:NameIDFormat>
    <md:SingleSignOnService Binding="urn:oasis:names:tc:SAML:2.0:bindings:HTTP-
POST" Location="idp-sso-url"/>
```

```
    <md:SingleSignOnService Binding="urn:oasis:names:tc:SAML:2.0:bindings:HTTP-
Redirect" Location="idp-sso-url"/>
  </md:IDPSSODescriptor>
</md:EntityDescriptor>
```

10. Biarkan bidang atribut Custom user ID kosong untuk menggunakan `NameID` elemen pernyataan SAMP untuk nama pengguna. Jika penegasan Anda tidak menggunakan elemen standar ini dan sebagai gantinya menyertakan nama pengguna sebagai atribut kustom, tentukan atribut di sini. Atribut peka huruf besar/kecil. Hanya satu atribut pengguna yang didukung.

    Contoh berikut menunjukkan atribut override untuk `NameID` dalam pernyataan SAMP:

    ```
    <saml2:Attribute Name="UserId" NameFormat="urn:oasis:names:tc:SAML:2.0:attrname-
    format:basic">
      <saml2:AttributeValue xmlns:xs="http://www.w3.org/2001/XMLSchema"
      xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
      xsi:type="xs:string">annie</saml2:AttributeValue>
    </saml2:Attribute>
    ```

11. (Opsional) Tentukan atribut kustom di bidang atribut Grup, seperti `role` atau`group`. Hanya satu atribut grup yang didukung. Tidak ada atribut grup default. Jika Anda tidak menentukannya, kebijakan akses data Anda hanya dapat berisi prinsipal pengguna.

    Contoh berikut menunjukkan atribut grup dalam pernyataan SAMP:

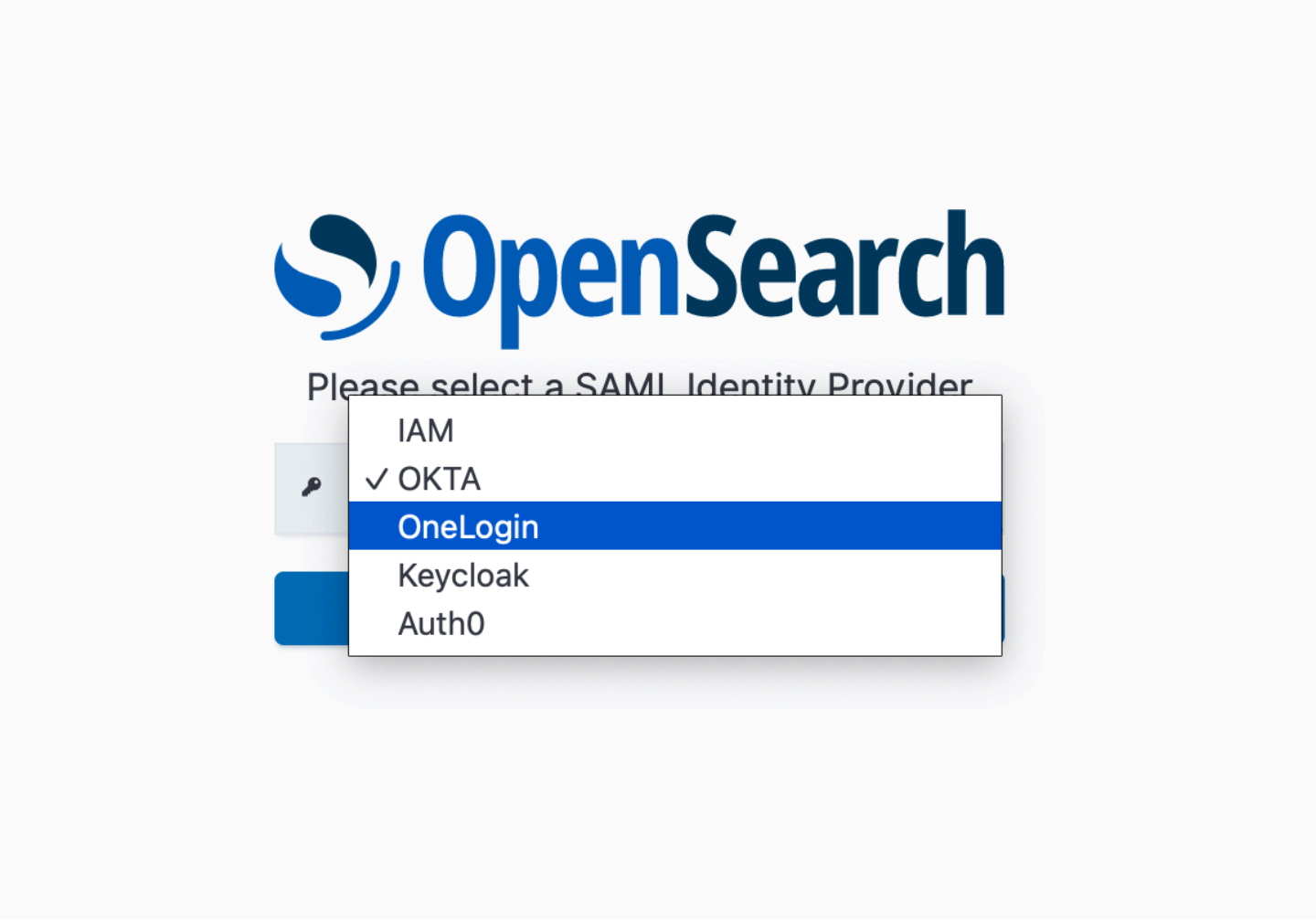
    ```
    <saml2:Attribute Name="department"
     NameFormat="urn:oasis:names:tc:SAML:2.0:attrname-format:basic">
        <saml2:AttributeValue xmlns:xs="http://www.w3.org/2001/XMLSchema"
        xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
        xsi:type="xs:string">finance</saml2:AttributeValue>
    </saml2:Attribute>
    ```

12. Secara default, OpenSearch Dasbor membuat pengguna keluar setelah 24 jam. Anda dapat mengonfigurasi nilai ini ke angka apa pun antara 1 dan 12 jam (15 dan 720 menit) dengan menentukan batas waktu OpenSearch Dasbor. Jika Anda mencoba mengatur batas waktu sama dengan atau kurang dari 15 menit, sesi Anda akan diatur ulang menjadi satu jam.
13. Pilih Buat penyedia SAMP.

Mengakses Dasbor OpenSearch

Setelah Anda mengonfigurasi penyedia SAMP, semua pengguna dan grup yang terkait dengan penyedia tersebut dapat menavigasi ke titik akhir OpenSearch Dasbor. URL Dasbor memiliki format *collection-endpoint*`/_dashboards/` untuk semua koleksi.

Jika Anda mengaktifkan SAMP, memilih tautan di AWS Management Console mengarahkan Anda ke halaman pilihan iDP, tempat Anda dapat masuk menggunakan kredensyal SAMP Anda. Pertama, gunakan dropdown untuk memilih penyedia identitas:



Kemudian masuk menggunakan kredensyal iDP Anda.

Jika Anda tidak mengaktifkan SAMP, memilih tautan di AWS Management Console mengarahkan Anda untuk masuk sebagai pengguna atau peran IAM, tanpa opsi untuk SAMP.

Memberikan akses identitas SAMP ke pengumpulan data

Setelah membuat penyedia SAMP, Anda masih perlu memberi pengguna dan grup yang mendasarinya akses ke data dalam koleksi Anda. Anda memberikan akses melalui kebijakan akses data. Sampai Anda memberikan akses kepada pengguna, mereka tidak akan dapat membaca, menulis, atau menghapus data apa pun dalam koleksi Anda.

Untuk memberikan akses, buat kebijakan akses data dan tentukan ID pengguna dan/atau grup SAMP Anda dalam `Principal` pernyataan:

```
[
   {
      "Rules":[
       ...
      ],
      "Principal":[
         "saml/987654321098/myprovider/user/Shaheen",
         "saml/987654321098/myprovider/group/finance"
      ]
   }
]
```

Anda dapat memberikan akses ke koleksi, indeks, atau keduanya. Jika Anda ingin pengguna yang berbeda memiliki izin yang berbeda, buat beberapa aturan. Untuk daftar izin yang tersedia, lihat Izin kebijakan yang didukung. Untuk informasi tentang cara memformat kebijakan akses, lihat Sintaks kebijakan.

Membuat penyedia SAMP ()AWS CLI

Untuk membuat penyedia SAMP menggunakan API OpenSearch Tanpa Server, kirim permintaan: CreateSecurityConfig

```
aws opensearchserverless create-security-config \
    --name myprovider \
    --type saml \
    --saml-options file://saml-auth0.json
```

Tentukan`saml-options`, termasuk metadata XML, sebagai peta nilai kunci dalam file.json. Metadata XMLharus dikodekan sebagai string lolos JSON.

```
{
```

```
   "sessionTimeout": 70,
   "groupAttribute": "department",
   "userAttribute": "userid",
   "metadata": "<EntityDescriptor xmlns=\"urn:oasis:names:tc:SAML:2.0:metadata
\" ... ... ... IDPSSODescriptor>\r\n<\/EntityDescriptor>"
}
```

Melihat penyedia SAMP

ListSecurityConfigsPermintaan berikut mencantumkan semua penyedia SAMP di akun Anda:

```
aws opensearchserverless list-security-configs --type saml
```

Permintaan mengembalikan informasi tentang semua penyedia SAMP yang ada, termasuk metadata IDP lengkap yang dihasilkan oleh penyedia identitas Anda:

```
{
   "securityConfigDetails": [
      {
         "configVersion": "MTY2NDA1MjY4NDQ5M18x",
         "createdDate": 1664054180858,
         "description": "Example SAML provider",
         "id": "saml/123456789012/myprovider",
         "lastModifiedDate": 1664054180858,
         "samlOptions": {
            "groupAttribute": "department",
            "metadata": "<EntityDescriptor xmlns=\"urn:oasis:names:tc:SAML:2.0:metadata
\" ... ... ... IDPSSODescriptor>\r\n<\/EntityDescriptor>",
            "sessionTimeout": 120,
            "userAttribute": "userid"
         }
      }
   ]
}
```

Untuk melihat detail tentang penyedia tertentu, termasuk pembaruan `configVersion` for future, kirim `GetSecurityConfig` permintaan.

Memperbarui penyedia SAMP

Untuk memperbarui penyedia SAMP menggunakan konsol OpenSearch Tanpa Server, pilih otentikasi SAMP, pilih penyedia identitas Anda, dan pilih Edit. Anda dapat memodifikasi semua bidang, termasuk metadata dan atribut kustom.

Untuk memperbarui penyedia melalui API OpenSearch Tanpa Server, kirim UpdateSecurityConfigpermintaan dan sertakan pengenal kebijakan yang akan diperbarui. Anda juga harus menyertakan versi konfigurasi, yang dapat Anda ambil menggunakan `GetSecurityConfig` perintah `ListSecurityConfigs` atau. Menyertakan versi terbaru memastikan bahwa Anda tidak secara tidak sengaja mengesampingkan perubahan yang dilakukan oleh orang lain.

Permintaan berikut memperbarui opsi SAMP untuk penyedia:

```
aws opensearchserverless update-security-config \
    --id saml/123456789012/myprovider \
    --type saml \
    --saml-options file://saml-auth0.json \
    --config-version MTY2NDA1MjY4NDQ5M18x
```

Tentukan opsi konfigurasi SAMP Anda sebagai peta nilai kunci dalam file.json.

> **Important**
>
> Pembaruan untuk opsi SAMP tidak inkremental. Jika Anda tidak menentukan nilai untuk parameter dalam `SAMLOptions` objek saat Anda melakukan pembaruan, nilai yang ada akan diganti dengan nilai kosong. Misalnya, jika konfigurasi saat ini berisi nilai untuk`userAttribute`, dan kemudian Anda membuat pembaruan dan tidak menyertakan nilai ini, nilai akan dihapus dari konfigurasi. Pastikan Anda tahu apa nilai yang ada sebelum Anda membuat pembaruan dengan memanggil `GetSecurityConfig` operasi.

Menghapus penyedia SAMP

Saat Anda menghapus penyedia SAMP, referensi apa pun ke pengguna dan grup terkait dalam kebijakan akses data Anda tidak lagi berfungsi. Untuk menghindari kebingungan, kami sarankan Anda menghapus semua referensi ke titik akhir dalam kebijakan akses Anda sebelum Anda menghapus titik akhir.

Untuk menghapus penyedia SAMP menggunakan konsol OpenSearch Tanpa Server, pilih Autentikasi, pilih penyedia, dan pilih Hapus.

Untuk menghapus penyedia melalui API OpenSearch Tanpa Server, kirim permintaan: DeleteSecurityConfig

```
aws opensearchserverless delete-security-config --id saml/123456789012/myprovider
```

Validasi kepatuhan untuk Amazon Tanpa Server OpenSearch

Auditor pihak ketiga menilai keamanan dan kepatuhan Amazon OpenSearch Serverless sebagai bagian dari beberapa AWS program kepatuhan. Hal ini mencakup SOC, PCI, dan HIPAA.

Untuk mempelajari apakah an Layanan AWS berada dalam lingkup program kepatuhan khusus, lihat Layanan AWS di Scope oleh Program Program Kepatuhan yang Anda minati. Untuk informasi umum, silakan lihat Program Kepatuhan AWS .

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

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

- Panduan Quick Start Keamanan dan 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.
- Merancang Keamanan dan Kepatuhan HIPAA di Amazon Web Services – Laporan resmi ini menjelaskan cara perusahaan dapat menggunakan AWS untuk membuat aplikasi yang memenuhi syarat HIPAA.

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

- Sumber Daya Kepatuhan AWS – Kumpulan buku kerja dan panduan ini mungkin berlaku untuk industri dan lokasi Anda.

- Panduan Kepatuhan Pelanggan AWS – Pahami model tanggung jawab bersama melalui lensa kepatuhan. Panduan ini merangkum praktik terbaik untuk mengamankan Layanan AWS dan memetakan panduan kontrol keamanan di banyak kerangka kerja (termasuk National Institute of Standards and Technology (NIST), Dewan Standar Keamanan Industri Kartu Pembayaran (PCI), dan Organisasi Internasional untuk Standardisasi (ISO)).
- Mengevaluasi Sumber Daya dengan Aturan di Panduan Developer AWS Config – Layanan AWS Config menilai seberapa baik konfigurasi sumber daya Anda dalam mematuhi praktik-praktik internal, pedoman industri, dan regulasi internal.
- AWS Security Hub – Layanan AWS ini memberikan pandangan komprehensif tentang status keamanan Anda di dadlam AWS. Security Hub menggunakan kontrol keamanan untuk sumber daya AWS Anda dan untuk memeriksa kepatuhan terhadap standar industri keamanan dan praktik terbaik. Untuk daftar layanan dan kontrol yang didukung, lihat Referensi kontrol Security Hub.
- AWS Audit Manager – Layanan AWS ini akan membantu Anda untuk terus-menerus mengaudit penggunaan AWS untuk menyederhanakan bagaimana Anda mengelola risiko dan kepatuhan terhadap regulasi dan standar industri.

Menandai koleksi Amazon OpenSearch Tanpa Server

Tanda memungkinkan Anda menetapkan informasi arbitrer ke koleksi Amazon OpenSearch Tanpa Server sehingga Anda dapat mengkategorikan dan mem-filter informasi tersebut. Tanda adalah label metadata yang Anda tetapkan atau AWS yang ditetapkan ke sumber daya AWS.

Setiap tanda terdiri dari kunci dan nilai. Untuk tanda yang Anda tetapkan, Anda menentukan kunci dan nilai. Misalnya, Anda dapat menentukan kunci sebagai `stage` dan nilai untuk satu sumber daya sebagai `test`.

Dengan tanda, Anda dapat melakukan hal berikut:

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

Dalam OpenSearch Tanpa Server, sumber daya utama adalah koleksi. Anda dapat menggunakan konsol OpenSearch Layanan, operasi API OpenSearch Tanpa Server, atauAWS SDK untuk menambah, mengelola, dan menghapus tag dari koleksi.AWS CLI

Izin diperlukan

OpenSearch Tanpa server menggunakan izinAWS Identity and Access Management Access Analyzer (IAM) berikut untuk penandaan koleksi:

- `aoss:TagResource`
- `aoss:ListTagsForResource`
- `aoss:UntagResource`

Cara menggunakan tanda (konsol)

Konsol adalah cara termudah untuk menandai koleksi.

Membuat tag (konsol)

1. Masuk ke konsol Amazon OpenSearch Service di https://console.aws.amazon.com/aos/home.
2. Perluas Tanpa Server di panel navigasi kiri dan pilih Koleksi.
3. Pilih koleksi yang ingin Anda tambahkan tanda, dan buka tab Tanda.
4. Pilih Kelola dan Tambahkan tag baru.
5. Masukkan kunci tanda dan nilai opsional.
6. Pilih Save (Simpan).

Untuk menghapus tag, ikuti langkah yang sama dan pilih Hapus di halaman Kelola tag.

Untuk informasi selengkapnya tentang menggunakan konsol untuk bekerja dengan tanda, lihat Tag Editor dalam Panduan Memulai KonsolAWS Manajemen.

Cara menggunakan tanda (AWS CLI)

Untuk menandai koleksi menggunakanAWS CLI, kirim TagResourcepermintaan:

```
aws opensearchserverless tag-resource
```

```
  --resource-arn arn:aws:aoss:us-east-1:123456789012:collection/my-collection
  --tags Key=service,Value=aoss Key=source,Value=logs
```

Melihat tag yang ada untuk koleksi dengan ListTagsForResourceperintah:

```
aws opensearchserverless list-tags-for-resource
  --resource-arn arn:aws:aoss:us-east-1:123456789012:collection/my-collection
```

Hapus tag dari koleksi menggunakan UntagResourceperintah:

```
aws opensearchserverless untag-resource
  --resource-arn arn:aws:aoss:us-east-1:123456789012:collection/my-collection
  --tag-keys service
```

Operasi dan plugin yang didukung di Amazon Tanpa Server OpenSearch

Amazon OpenSearch Serverless mendukung berbagai OpenSearch plugin, serta subset dari operasi API pengindeksan, pencarian, dan metadata yang tersedia di. OpenSearch Anda dapat menyertakan izin di kolom kiri tabel dalam kebijakan akses data untuk membatasi akses ke operasi tertentu.

Topik

- Operasi dan izin OpenSearch API yang didukung
- OpenSearch Plugin yang didukung

Operasi dan izin OpenSearch API yang didukung

Tabel berikut mencantumkan operasi API yang didukung OpenSearch Tanpa Server, bersama dengan izin IAM yang sesuai:

Izin kebijakan akses data	OpenSearch Operasi API	Deskripsi dan peringatan
`aoss:CreateIndex`	MENEMPATKAN <index>	Buat indeks. Untuk informasi selengkapnya, lihat Membuat indeks.

Izin kebijakan akses data	OpenSearch Operasi API	Deskripsi dan peringatan
		Note Izin ini juga berlaku untuk membuat indeks dengan data sampel di OpenSearch Dasbor.
`aoss:DescribeIndex`	• DAPATKAN <index> • DAPATKAN <index>/_mapping • DAPATKAN <index>/_mappings • DAPATKAN <index>/_setting • DAPATKAN <index>/_setting/<setting> • DAPATKAN <index>/_settings • DAPATKAN <index>/_settings/<setting> • DAPATKAN _kucing/indeks • DAPATKAN _pemetaan • DAPATKAN _pemetaan • DAPATKAN _selesaikan/indeks/<index>	Jelaskan indeks. Untuk informasi selengkapnya, lihat sumber daya berikut: • Dapatkan indeks • Dapatkan pemetaan • Dapatkan pengaturan • Indeks CAT (Respons tidak termasuk `health` atau `status` bidang.)

Izin kebijakan akses data	OpenSearch Operasi API	Deskripsi dan peringatan
`aoss:WriteDocument`	• <index>HAPUS/_doc/ <id> • POSTING <index>/_bulk • POST <index>/_create/ <id>(hanya untuk jenis koleksi pencarian) • POSTING <index>/_doc • POSTING <index>/_update/ <id> • POST _massal • PUT <index>/_create/ <id>(hanya untuk jenis koleksi pencarian) • PUT <index>/_doc/ <id>(hanya untuk jenis koleksi pencarian)	Tulis dan perbarui dokumen. Untuk informasi selengkapnya, lihat sumber daya berikut: • Massal • Data indeks **Note** Beberapa operasi hanya diperbolehkan untuk koleksi jenis`SEARCH`. Untuk informasi selengkapnya, lihat the section called “Memilih jenis koleksi”.

Izin kebijakan akses data	OpenSearch Operasi API	Deskripsi dan peringatan
`aoss:ReadDocument`	• DAPATKAN <index>/_analisis • DAPATKAN <index>/_doc/ <id> • DAPATKAN <index>/_jelaskan/ <id> • DAPATKAN <index>/_mget • DAPATKAN <index>/_source/ <id> • DAPATKAN <index>/_hitung • DAPATKAN <index>/_field_caps • DAPATKAN <index>/_msearch • DAPATKAN <index>/_rank_eval • DAPATKAN <index>/_search • DAPATKAN <index>/_validate/ <query> • DAPATKAN _analisis • DAPATKAN _field_caps • DAPATKAN _mget • DAPATKAN _cari • KEPALA <index>/_doc/ <id> • KEPALA <index>/_sumber/ <id> • POST <index>/_analisis • POSTING <index>/_menjelaskan/ <id> • POSTING <index>/_hitung • POSTING <index>/_field_caps • POST <index>/_rank_eval • POST <index>/_search • POST _analisis • POSTING _field_caps	Baca dokumen. Untuk informasi selengkapnya, lihat sumber daya berikut: • Lakukan analisis teks • Dapatkan dokumen • Hitungan • Kueri DSL • Evaluasi peringkat • Analisis API • Jelaskan

Izin kebijakan akses data	OpenSearch Operasi API	Deskripsi dan peringatan
	• POST _pencarian	
`aoss:DeleteIndex`	HAPUS <target>	Hapus indeks. Untuk informasi selengkapnya, lihat Menghapus indeks.
`aoss:UpdateIndex`	• Pemetaan POST • POSTING <index>/_pemetaan/ • POSTING <index>/_pemetaan/ • POST <index>/_setting • POST <index>/_settings • POST _pengaturan • POST _pengaturan • PUT _pemetaan • PUT <index>/_mapping • <index>MASUKKAN/_mappings/ • <index>MASUKKAN/_setting • <index>MASUKKAN/_settings • PUT _pengaturan • PUT _pengaturan	Perbarui pengaturan indeks. Untuk informasi selengkapnya, lihat sumber daya berikut: • Pemetaan • Perbarui pengaturan
`aoss:CreateCollectionItems`	_alias POST	Buat alias indeks. Untuk informasi selengkapnya, lihat Membuat alias.

Izin kebijakan akses data	OpenSearch Operasi API	Deskripsi dan peringatan
`aoss:DescribeCollectionItems`	• DAPATKAN <index>/_alias/ <alias> • DAPATKAN _alias • DAPATKAN _alias/ <alias> • DAPATKAN _kucing/alias • DAPATKAN _kucing/templat • DAPATKAN _kucing/template/ <template_name> • DAPATKAN _component_template • DAPATKAN _component_template/ <component-template> • DAPATKAN _index_template • DAPATKAN _index_template/ <index-template> • KEPALA _alias/ <alias> • KEPALA _component_template/ <component-template> • KEPALA _index_template/ <name> • KEPALA <index>/_alias/<alias>	Jelaskan alias dan templat indeks. Untuk informasi selengkapnya, lihat sumber daya berikut: • Kelola alias • Template indeks

Izin kebijakan akses data	OpenSearch Operasi API	Deskripsi dan peringatan
`aoss:UpdateCollectionItems`	• POSTING <index>/_alias/ <alias> • POSTING <index>/_aliases/ <alias> • POSTING _component_template/ <component-template> • POSTING _index_template/ <index-template> • MASUKKAN/_alias<index>/<alias> • <index>MASUKKAN/_aliases/ <alias> • MASUKKAN _component_template/ <component-template> • MASUKKAN _index_template/ <index-template>	Perbarui alias dan templat indeks. Untuk informasi selengkapnya, lihat sumber daya berikut: • Alias indeks • Template indeks
`aoss:DeleteCollectionItems`	• <index>HAPUS/_alias/<alias> • HAPUS _component_template/ <component-template> • HAPUS _index_template/ <index-template> • <index>HAPUS/_aliases/ <alias>	Hapus alias dan templat indeks. Untuk informasi selengkapnya, lihat sumber daya berikut: • Hapus alias • Hapus template

OpenSearch Plugin yang didukung

OpenSearch Koleksi tanpa server dikemas dengan plugin berikut dari komunitas. OpenSearch Tanpa server secara otomatis menyebarkan dan mengelola plugin untuk Anda.

Plugin analisis

- Analisis ICU
- Analisis Jepang (kuromoji)

- Analisis Korea (Nori)
- Analisis Fonetik
- Analisis Bahasa Mandarin Cerdas
- Gambar Analisis Polandia
- Analisis Ukraina

Plugin Mapper

- Ukuran Mapper
- Pemetaan Murmur3
- Teks Beranotasi Mapper

Plugin skrip

- Tanpa rasa sakit
- Ekspresi
- Kumis

Selain itu, OpenSearch Serverless mencakup semua plugin yang dikirimkan sebagai modul.

Memantau Amazon Tanpa OpenSearch Server

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

- Amazon CloudWatch memantau AWS sumber daya Anda dan aplikasi yang Anda jalankan AWS secara real time. Anda dapat mengumpulkan dan melacak metrik, membuat dasbor yang disesuaikan, dan mengatur alarm yang memberi tahu Anda atau mengambil tindakan saat metrik tertentu mencapai ambang batas yang ditentukan.

  Misalnya, Anda dapat CloudWatch melacak penggunaan CPU atau metrik lain dari instans Amazon EC2 Anda dan secara otomatis meluncurkan instans baru bila diperlukan. Untuk informasi selengkapnya, lihat Panduan CloudWatch Pengguna Amazon.

- AWS CloudTrail menangkap panggilan API dan peristiwa terkait yang dilakukan oleh atau atas nama Akun AWSAnda. Ini mengirimkan file log ke bucket Amazon S3 yang Anda tentukan. Anda dapat mengidentifikasi pengguna dan akun mana yang dipanggil AWS, alamat IP sumber dari mana panggilan dilakukan, dan kapan panggilan terjadi. Untuk informasi selengkapnya, silakan lihat Panduan PenggunaAWS CloudTrail.
- Amazon EventBridge memberikan aliran kejadian sistem yang mendekati real-time yang menjelaskan perubahan dalam domain OpenSearch Layanan Anda. Anda dapat membuat aturan yang mengawasi peristiwa tertentu, dan memicu tindakan otomatis di tempat lain Layanan AWS saat peristiwa ini terjadi. Untuk informasi selengkapnya, lihat Panduan EventBridge Pengguna Amazon.

Memantau OpenSearch Tanpa Server dengan Amazon CloudWatch

Anda dapat memantau penggunaan Amazon OpenSearch Tanpa Server CloudWatch, yang mengumpulkan data mentah dan memprosesnya menjadi metrik yang dapat dibaca, mendekati waktu nyata. Statistik ini disimpan untuk jangka waktu 15 bulan, sehingga Anda dapat mengakses informasi historis dan mendapatkan perspektif yang lebih baik tentang performa aplikasi atau layanan web Anda.

Anda juga dapat mengatur alarm yang memperhatikan ambang batas tertentu dan mengirim notifikasi atau mengambil tindakan saat ambang batas tersebut terpenuhi. Untuk informasi selengkapnya, lihat Panduan CloudWatch Pengguna Amazon.

OpenSearch Tanpa server melaporkan metrik berikut di namespace. `AWS/AOSS`

Metrik	Deskripsi
`ActiveCollection`	Menunjukkan apakah koleksi aktif. Nilai 1 berarti bahwa koleksi berada dalam `ACTIVE` keadaan. Nilai ini dipancark an setelah berhasil membuat koleksi dan tetap 1 sampai Anda menghapus koleksi. Metrik tidak dapat memiliki nilai 0. Statistik yang relevan: Max Dimensi:`ClientId`,`CollectionId` , `Collectio nName`

Metrik	Deskripsi
	Frekuensi: 60 detik
`DeletedDocuments`	Jumlah total dokumen yang dihapus. Statistik yang relevan: Rata-rata, Jumlah Dimensi:`ClientId`,`CollectionId` ,`CollectionName` ,`IndexId`, `IndexName` Frekuensi: 60 detik
`IndexingOCU`	Jumlah OpenSearch Compute Units (OCU) yang digunakan untuk menelan data pengumpulan. Metrik ini berlaku di tingkat akun. Statistik yang relevan: Jumlah Dimensi: `ClientId` Frekuensi: 60 detik
`IngestionDataRate`	Tingkat pengindeksan dalam GiB per detik untuk koleksi atau indeks. Metrik ini hanya berlaku untuk permintaan pengindeksan massal. Statistik yang relevan: Jumlah Dimensi:`ClientId`,`CollectionId` ,`CollectionName` ,`IndexId`, `IndexName` Frekuensi: 60 detik

Metrik	Deskripsi
`IngestionDocumentErrors`	Jumlah total kesalahan dokumen selama konsumsi untuk koleksi atau indeks. Setelah permintaan pengindeksan massal berhasil, penulis memproses permintaan dan mengeluarkan kesalahan untuk semua dokumen yang gagal dalam permintaan. Statistik yang relevan: Jumlah Dimensi:`ClientId`,`CollectionId` ,`CollectionName` ,`IndexId`, `IndexName` Frekuensi: 60 detik
`IngestionDocumentRate`	Tingkat per detik di mana dokumen dicerna ke koleksi atau indeks. Metrik ini hanya berlaku untuk permintaan pengindeksan massal. Statistik yang relevan: Jumlah Dimensi:`ClientId`,`CollectionId` ,`CollectionName` ,`IndexId`, `IndexName` Frekuensi: 60 detik
`IngestionRequestErrors`	Jumlah total kesalahan permintaan pengindeksan massal ke koleksi. OpenSearch Tanpa server memancarkan metrik ini ketika permintaan pengindeksan massal gagal karena alasan apa pun, seperti masalah otentikasi atau ketersediaan. Statistik yang relevan: Jumlah Dimensi:`ClientId`,`CollectionId` , `CollectionName` Frekuensi: 60 detik

Metrik	Deskripsi
`IngestionRequestLatency`	Latensi, dalam hitungan detik, untuk operasi penulisan massal ke koleksi. Statistik yang relevan: Minimum, Maksimum, Rata-rata Dimensi:`ClientId`,`CollectionId` , `CollectionName` Frekuensi: 60 detik
`IngestionRequestRate`	Jumlah total operasi penulisan massal yang diterima oleh koleksi. Statistik yang relevan: Minimum, Maksimum, Rata-rata Dimensi:`ClientId`,`CollectionId` , `CollectionName` Frekuensi: 60 detik
`IngestionRequestSuccess`	Jumlah total operasi pengindeksan yang berhasil ke koleksi. Statistik yang relevan: Jumlah Dimensi:`ClientId`,`CollectionId` , `CollectionName` Frekuensi: 60 detik
`SearchableDocuments`	Jumlah total dokumen yang dapat dicari dalam koleksi atau indeks. Statistik yang relevan: Jumlah Dimensi:`ClientId`,`CollectionId` ,`CollectionName` ,`IndexId`, `IndexName` Frekuensi: 60 detik

Metrik	Deskripsi
`SearchRequestErrors`	Jumlah total kesalahan kueri per menit untuk koleksi. Statistik yang relevan: Jumlah Dimensi:`ClientId`,`CollectionId` , `CollectionName` Frekuensi: 60 detik
`SearchRequestLatency`	Waktu rata-rata, dalam milidetik, yang diperlukan untuk menyelesaikan operasi pencarian terhadap koleksi. Statistik yang relevan: Minimum, Maksimum, Rata-rata Dimensi:`ClientId`,`CollectionId` , `CollectionName` Frekuensi: 60 detik
`SearchOCU`	Jumlah OpenSearch Compute Units (OCU) yang digunakan untuk mencari data pengumpulan. Metrik ini berlaku di tingkat akun. Statistik yang relevan: Jumlah Dimensi: `ClientId` Frekuensi: 60 detik
`SearchRequestRate`	Jumlah total permintaan pencarian per menit untuk koleksi. Statistik yang relevan: Rata-rata, Maksimum, Jumlah Dimensi:`ClientId`,`CollectionId` , `CollectionName` Frekuensi: 60 detik

Metrik	Deskripsi
`StorageUsedInS3`	Jumlah, dalam byte, penyimpanan Amazon S3 yang digunakan. OpenSearch Tanpa server menyimpan data yang diindeks di Amazon S3. Anda harus memilih periode pada satu menit untuk mendapatkan nilai yang akurat. Statistik yang relevan: Jumlah Dimensi:`ClientId`,`CollectionId` ,`CollectionName` ,`IndexId`, `IndexName` Frekuensi: 60 detik
`2xx, 3xx, 4xx, 5xx`	Jumlah permintaan ke koleksi yang menghasilkan kode respons HTTP yang diberikan (2 xx, 3 xx, 4 xx, 5 xx). Statistik yang relevan: Jumlah Dimensi:`ClientId`,`CollectionId` , `CollectionName` Frekuensi: 60 detik

Mencatat OpenSearch panggilan API Tanpa Server menggunakan AWS CloudTrail

Amazon OpenSearch Serverless terintegrasi dengan AWS CloudTrail, layanan yang menyediakan catatan tindakan yang diambil oleh pengguna, peran, atau AWS layanan di Tanpa Server.

CloudTrail menangkap semua panggilan API untuk OpenSearch Tanpa Server sebagai peristiwa. Panggilan yang diambil mencakup panggilan dari bagian Tanpa Server pada konsol OpenSearch Layanan dan panggilan kode ke operasi API Tanpa OpenSearch Server.

Jika Anda membuat jejak, Anda dapat mengaktifkan pengiriman CloudTrail acara secara terus menerus ke bucket Amazon S3, termasuk peristiwa untuk OpenSearch Tanpa Server. Jika Anda tidak mengonfigurasi jejak, Anda masih dapat melihat peristiwa terbaru di CloudTrail konsol dalam Riwayat acara.

Dengan menggunakan informasi yang dikumpulkan oleh CloudTrail, Anda dapat menentukan permintaan yang dibuat untuk OpenSearch Tanpa Server, alamat IP dari mana permintaan dibuat, siapa yang membuat permintaan, kapan dibuat, dan detail tambahan.

Untuk mempelajari selengkapnya CloudTrail, lihat PanduanAWS CloudTrail Pengguna.

OpenSearch Informasi tanpa server di CloudTrail

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

Untuk catatan acara yang sedang berlangsung di Anda Akun AWS, termasuk acara untuk OpenSearch Tanpa Server, buat jejak. Jejak memungkinkan CloudTrail untuk mengirimkan file log ke bucket Amazon S3. Secara default, saat Anda membuat jejak di konsol, jejak tersebut berlaku untuk semua Wilayah AWS.

Jejak mencatat peristiwa dari semua Wilayah di AWS partisi dan mengirimkan file log ke bucket Amazon S3 yang Anda tentukan. Selain itu, Anda dapat mengonfigurasi AWS layanan lain untuk menganalisis lebih lanjut dan menindaklanjuti data peristiwa yang dikumpulkan dalam CloudTrail log. Untuk informasi selengkapnya, lihat berikut:

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

Semua tindakan OpenSearch Tanpa Server dicatat oleh CloudTrail dan didokumentasikan dalam referensi API Tanpa OpenSearch Server. Misalnya, panggilan ke`CreateCollection`,`ListCollections`, dan `DeleteCollection` tindakan menghasilkan entri dalam file CloudTrail log.

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

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

Untuk informasi selengkapnya, lihat Elemen userIdentity CloudTrail .

Memahami OpenSearch entri file log tanpa server

Trail adalah konfigurasi yang memungkinkan pengiriman peristiwa sebagai file log ke bucket Amazon S3 yang Anda tentukan. CloudTrail file log berisi satu atau lebih entri log.

Peristiwa menunjukkan satu permintaan dari sumber mana pun. Ini mencakup informasi tentang tindakan yang diminta, tanggal dan waktu tindakan, parameter permintaan, dan sebagainya. CloudTrail file log bukanlah jejak tumpukan yang diurutkan dari panggilan API publik, jadi file tersebut tidak muncul dalam urutan tertentu.

Contoh berikut menunjukkan entri CloudTrail log yang menunjukkan `CreateCollection` tindakan.

```
{
   "eventVersion":"1.08",
   "userIdentity":{
      "type":"AssumedRole",
      "principalId":"AIDACKCEVSQ6C2EXAMPLE",
      "arn":"arn:aws:iam::123456789012:user/test-user",
      "accountId":"123456789012",
      "accessKeyId":"access-key",
      "sessionContext":{
         "sessionIssuer":{
            "type":"Role",
            "principalId":"AIDACKCEVSQ6C2EXAMPLE",
            "arn":"arn:aws:iam::123456789012:role/Admin",
            "accountId":"123456789012",
            "userName":"Admin"
         },
         "webIdFederationData":{

         },
         "attributes":{
```

```
            "creationDate":"2022-04-08T14:11:34Z",
            "mfaAuthenticated":"false"
         }
      }
   },
   "eventTime":"2022-04-08T14:11:49Z",
   "eventSource":"aoss.amazonaws.com",
   "eventName":"CreateCollection",
   "awsRegion":"us-east-1",
   "sourceIPAddress":"AWS Internal",
   "userAgent":"aws-cli/2.1.30 Python/3.8.8 Linux/5.4.176-103.347.amzn2int.x86_64 exe/
x86_64.amzn.2 prompt/off command/aoss.create-collection",
   "errorCode":"HttpFailureException",
   "errorMessage":"An unknown error occurred",
   "requestParameters":{
      "accountId":"123456789012",
      "name":"test-collection",
      "description":"A sample collection",
      "clientToken":"d3a227d2-a2a7-49a6-8fb2-e5c8303c0718"
   },
   "responseElements": null,
   "requestID":"12345678-1234-1234-1234-987654321098",
   "eventID":"12345678-1234-1234-1234-987654321098",
   "readOnly":false,
   "eventType":"AwsApiCall",
   "managementEvent":true,
   "recipientAccountId":"123456789012",
   "eventCategory":"Management",
   "tlsDetails":{
      "clientProvidedHostHeader":"user.aoss-sample.us-east-1.amazonaws.com"
   }
}
```


Memantau peristiwa OpenSearch Tanpa Server menggunakan Amazon EventBridge

OpenSearch Layanan Amazon terintegrasi dengan Amazon EventBridge untuk memberi tahu Anda tentang peristiwa tertentu yang memengaruhi domain Anda. Acara dari AWS layanan dikirimkan ke EventBridge dalam waktu dekat. Acara yang sama juga dikirim ke Amazon CloudWatch Events, pendahulu Amazon EventBridge. Anda dapat menulis aturan untuk menunjukkan acara mana yang menarik bagi Anda, dan tindakan otomatis apa yang harus diambil ketika suatu acara cocok dengan aturan. Contoh tindakan yang dapat Anda aktifkan secara otomatis meliputi yang berikut:

- Memanggil fungsi AWS Lambda
- Melakukan invokasi Amazon EC2 Run Command
- Mengirimkan kejadian ke Amazon Kinesis Data Streams
- Mengaktifkan mesin status AWS Step Functions
- Memberi tahu topik Amazon SNS atau antrean Amazon SQS

Untuk informasi selengkapnya, lihat Memulai Amazon EventBridge di Panduan EventBridge Pengguna Amazon.

Menyiapkan notifikasi

Anda dapat menggunakan PemberitahuanAWS Pengguna untuk menerima pemberitahuan saat peristiwa OpenSearch Tanpa Server terjadi. Peristiwa adalah indikator perubahan lingkungan OpenSearch Tanpa Server, seperti ketika Anda mencapai batas maksimum penggunaan OCU Anda. Amazon EventBridge menerima acara dan merutekan pemberitahuan ke Pusat AWS Management Console Pemberitahuan dan saluran pengiriman yang Anda pilih. Anda akan menerima notifikasi saat ada sebuah peristiwa yang cocok dengan sebuah aturan yang Anda tentukan.

OpenSearch Acara Compute Units (OCU)

OpenSearch Tanpa server mengirimkan peristiwa ke EventBridge saat salah satu peristiwa terkait OCU berikut terjadi.

Penggunaan OCU mendekati batas maksimum

OpenSearch Tanpa server mengirimkan acara ini ketika penggunaan OCU pencarian atau indeks Anda mencapai 75% dari batas kapasitas Anda. Penggunaan OCU Anda dihitung berdasarkan batas kapasitas yang dikonfigurasi dan konsumsi OCU Anda saat ini.

Contoh

Berikut ini adalah contoh peristiwa jenis ini (cari OCU):

```
{
  "version": "0",
  "id": "01234567-0123-0123-0123-012345678901",
  "detail-type": "OCU Utilization Approaching Max Limit",
  "source": "aws.aoss",
  "account": "123456789012",
```

```
  "time": "2016-11-01T13:12:22Z",
  "region": "us-east-1",
  "resources": ["arn:aws:es:us-east-1:123456789012:domain/test-domain"],
  "detail": {
    "eventTime" : 1678943345789,
    "description": "Your search OCU usage is at 75% and is approaching the configured
 maximum limit."
  }
}
```

Berikut ini adalah contoh peristiwa jenis ini (indeks OCU):

```
{
  "version": "0",
  "id": "01234567-0123-0123-0123-012345678901",
  "detail-type": "OCU Utilization Approaching Max Limit",
  "source": "aws.aoss",
  "account": "123456789012",
  "time": "2016-11-01T13:12:22Z",
  "region": "us-east-1",
  "resources": ["arn:aws:es:us-east-1:123456789012:domain/test-domain"],
  "detail": {
    "eventTime" : 1678943345789,
    "description": "Your indexing OCU usage is at 75% and is approaching the configured
 maximum limit."
  }
```

Penggunaan OCU mencapai batas maksimum

OpenSearch Tanpa server mengirimkan acara ini ketika penggunaan OCU pencarian atau indeks Anda mencapai 100% dari batas kapasitas Anda. Penggunaan OCU Anda dihitung berdasarkan batas kapasitas yang dikonfigurasi dan konsumsi OCU Anda saat ini.

Contoh

Berikut ini adalah contoh peristiwa jenis ini (cari OCU):

```
{
  "version": "0",
  "id": "01234567-0123-0123-0123-012345678901",
  "detail-type": "OCU Utilization Reached Max Limit",
```

```
  "source": "aws.aoss",
  "account": "123456789012",
  "time": "2016-11-01T13:12:22Z",
  "region": "us-east-1",
  "resources": ["arn:aws:es:us-east-1:123456789012:domain/test-domain"],
  "detail": {
    "eventTime" : 1678943345789,
    "description": "Your search OCU usage has reached the configured maximum limit."
  }
}
```

Berikut ini adalah contoh peristiwa jenis ini (indeks OCU):

```
{
  "version": "0",
  "id": "01234567-0123-0123-0123-012345678901",
  "detail-type": "OCU Utilization Reached Max Limit",
  "source": "aws.aoss",
  "account": "123456789012",
  "time": "2016-11-01T13:12:22Z",
  "region": "us-east-1",
  "resources": ["arn:aws:es:us-east-1:123456789012:domain/test-domain"],
  "detail": {
    "eventTime" : 1678943345789,
    "description": "Your indexing OCU usage has reached the configured maximum limit."
  }
}
```

Membuat dan mengelola domain OpenSearch Layanan Amazon

Bab ini menjelaskan cara membuat dan mengelola domain OpenSearch Layanan Amazon. Domain OpenSearch layanan identik dengan cluster. OpenSearch Domain adalah klaster dengan pengaturan, tipe instans, jumlah instans, dan sumber daya penyimpanan yang Anda tentukan.

Berbeda dengan instruksi singkat dalam tutorial Memulai, Bab ini menjelaskan semua opsi dan memberikan informasi referensi yang relevan. Anda dapat menyelesaikan setiap prosedur dengan menggunakan instruksi untuk konsol OpenSearch Layanan, AWS Command Line Interface (AWS CLI), atau AWS SDK.

Membuat domain OpenSearch Layanan

Bagian ini menjelaskan cara membuat domain OpenSearch Layanan dengan menggunakan konsol OpenSearch Layanan atau dengan menggunakan `create-domain` perintah AWS CLI with.

Membuat domain OpenSearch Layanan (konsol)

Gunakan prosedur berikut untuk membuat domain OpenSearch Layanan menggunakan konsol.

Untuk membuat domain OpenSearch Layanan (konsol)

1. Masuk ke https://aws.amazon.com, dan pilih Masuk ke Konsol.
2. Di bawah Analytics, pilih OpenSearch Layanan Amazon.
3. Pilih Create domain (Buat domain).
4. Untuk nama Domain, masukkan nama domain. Versi harus memenuhi kriteria berikut:
   - Unik untuk akun Anda dan Wilayah AWS
   - Dimulai dengan huruf kecil
   - Berisi antara 3 dan 28 karakter
   - Hanya berisi huruf kecil a-z, angka 0-9, dan tanda hubung (-)
5. Untuk metode pembuatan domain, pilih Standard create.
6. Untuk Template, pilih opsi yang paling sesuai dengan tujuan domain Anda:

- Domain produksi untuk beban kerja yang membutuhkan ketersediaan dan kinerja tinggi. Domain ini menggunakan Multi-AZ (dengan atau tanpa siaga) dan node master khusus untuk ketersediaan yang lebih tinggi.
- Dev/test untuk pengembangan atau pengujian. Domain ini dapat menggunakan Multi-AZ (dengan atau tanpa siaga) atau satu Availability Zone.

> **Important**
>
> Jenis deployment yang berbeda menyajikan pilihan yang berbeda pada halaman berikutnya. Langkah-langkah ini mencakup semua opsi.

7. Untuk Opsi Deployment, pilih Domain dengan standby untuk mengonfigurasi domain 3-AZ, dengan node di salah satu zona dicadangkan sebagai siaga. Opsi ini memberlakukan sejumlah praktik terbaik, seperti jumlah node data tertentu, jumlah node master, jenis instance, jumlah replika, dan pengaturan pembaruan perangkat lunak.
8. Untuk Versi, pilih versi OpenSearch atau lama Elasticsearch OSS yang akan digunakan. Kami menyarankan Anda memilih versi terbaru dari OpenSearch. Untuk informasi selengkapnya, lihat the section called “Versi OpenSearch dan Elasticsearch yang didukung”.

   (Opsional) Jika Anda memilih OpenSearch versi untuk domain Anda, pilih Aktifkan mode kompatibilitas untuk membuat OpenSearch laporkan versinya sebagai 7.10, yang memungkinkan klien dan plugin OSS Elasticsearch tertentu yang memeriksa versi sebelum menghubungkan untuk terus bekerja dengan layanan.
9. Untuk tipe Instance, pilih tipe instance untuk node data Anda. Untuk informasi selengkapnya, lihat the section called “Tipe instans yang didukung”.

   > **Note**
   >
   > Tidak semua Availability Zone mendukung semua tipe instans. Jika Anda memilih Multi-AZ dengan atau tanpa Standby, sebaiknya pilih jenis instans generasi saat ini, seperti R5 atau I3.
10. Untuk Jumlah simpul, pilih jumlah simpul data.

    Untuk nilai maksimum, lihat Domain OpenSearch layanan dan kuota instance. Klaster simpul tunggal bisa saja untuk pengembangan dan pengujian, tetapi tidak boleh digunakan untuk beban

kerja produksi. Untuk panduan lebih lanjut, lihat the section called “Mengukur domain” dan the section called “Mengonfigurasi domain Multi-AZ”.

11. Untuk jenis Penyimpanan, pilih Amazon EBS. Tipe volume yang tersedia di daftar tergantung pada tipe instans yang telah Anda pilih. Untuk panduan tentang membuat domain yang sangat besar, lihatthe section called “Menskalakan Petabyte”.
12. Untuk penyimpanan EBS, konfigurasikan pengaturan tambahan berikut. Beberapa pengaturan mungkin tidak muncul tergantung pada jenis volume yang Anda pilih.

Pengaturan	Deskripsi
Jenis volume EBS	Pilih antara General Purpose (SSD) - gp3 dan General Purpose (SSD) - gp2, atau IOPS Provisioned generasi sebelumnya (SSD), dan Magnetic (standar).
Ukuran penyimpanan EBS per node	Masukkan ukuran volume EBS yang ingin Anda lampirkan ke setiap node data. Ukuran volume EBS adalah per simpul. Anda dapat menghitung ukuran cluster total untuk domain OpenSearch Service dengan mengalikan jumlah node data dengan ukuran volume EBS. Ukuran minimum dan maksimum volume EBS tergantung pada tipe volume EBS yang ditentukan dan tipe instans yang dilampirkan. Untuk mempelajari lebih lanjut, lihat Batas ukuran volume EBS.
IOPS yang Tersedia	Jika Anda memilih jenis volume SSD IOPS Provisioned, masukkan jumlah operasi I/O per detik (IOPS) yang dapat didukung volume.

13. (Opsional) Jika Anda memilih jenis gp3 volume, perluas Pengaturan lanjutan dan tentukan IOPS tambahan (hingga 1.000 MiB/s untuk setiap ukuran volume 3 TiB yang disediakan per node data) dan throughput (hingga 16.000 untuk setiap ukuran volume 3 TiB yang disediakan per node data) untuk penyediaan setiap node, di luar yang disertakan dengan harga penyimpanan, dengan biaya tambahan. Untuk informasi selengkapnya, lihat harga OpenSearch Layanan Amazon.

14. (Opsional) Untuk mengaktifkan UltraWarm penyimpanan, pilih Aktifkan node UltraWarm data. Setiap tipe instans memiliki jumlah penyimpanan maksimum yang dapat diatasi. Kalikan jumlah itu dengan jumlah simpul data hangat untuk total penyimpanan hangat yang dapat dialamatkan.
15. (Opsional) Untuk mengaktifkan Penyimpanan dingin, pilih Aktifkan penyimpanan dingin. Anda harus mengaktifkan UltraWarm untuk mengaktifkan penyimpanan dingin.
16. Jika Anda menggunakan Multi-AZ dengan Standby, tiga node master khusus akan diaktifkan. Pilih jenis node master yang Anda inginkan. Jika Anda memilih domain Multi-AZ tanpa Standby, pilih Aktifkan node master khusus dan pilih jenis dan jumlah node master yang Anda inginkan. Simpul utama khusus meningkatkan stabilitas klaster dan diperlukan untuk domain yang memiliki jumlah instans lebih dari 10. Kami merekomendasikan tiga simpul utama khusus untuk domain produksi.

> **Note**
>
> Anda dapat memilih tipe instans yang berbeda untuk simpul utama dan simpul data khusus Anda. Misalnya, Anda dapat memilih instans tujuan umum atau penyimpanan yang dioptimalkan untuk node data Anda, tetapi contoh komputasi yang dioptimalkan untuk simpul utama khusus Anda.

17. (Opsional) Untuk domain yang berjalan OpenSearch atau Elasticsearch 5.3 dan yang lebih baru, konfigurasi Snapshot tidak relevan. Untuk informasi lebih lanjut tentang snapshot otomatis, lihat the section called “Membuat snapshot indeks”.
18. Jika Anda ingin menggunakan titik akhir kustom daripada yang standar`https://search-`*`mydomain`*`-`*`1a2a3a4a5a6a7a8a9a0a9a8a7a`*`.`*`us-east-1`*`.es.amazonaws.com`, pilih Aktifkan titik akhir kustom dan berikan nama dan sertifikat. Untuk informasi selengkapnya, lihat the section called “Membuat titik akhir kustom”.
19. Di bawah Jaringan, pilih akses VPC atau Akses publik. Jika Anda memilih Akses publik, lewati ke langkah berikutnya. Jika Anda memilih akses VPC, pastikan Anda memenuhi prasyarat, lalu konfigurasikan pengaturan berikut:

Pengaturan	Deskripsi
VPC	Pilih ID virtual private cloud (VPC) yang ingin Anda gunakan. VPC dan domain harus sama Wilayah AWS, dan Anda harus memilih VPC dengan penyewaan disetel ke Default. OpenSearch Layanan belum mendukung VPC yang menggunakan penyewaan khusus.

Pengaturan	Deskripsi
Subnet	Pilih subnet. Jika Anda mengaktifkan Multi-AZ, Anda harus memilih dua atau tiga subnet. OpenSearch Layanan akan menempatkan titik akhir VPC dan antarmuka jaringan elastis di subnet. Anda harus memesan alamat IP yang cukup untuk antarmuka jaringan di subnet. Untuk informasi selengkapnya, lihat Memesan alamat IP dalam subnet VPC.
Grup keamanan	Pilih satu atau beberapa grup keamanan VPC yang memungkinkan aplikasi Anda untuk mencapai domain OpenSearch Layanan pada port (80 atau 443) dan protokol (HTTP atau HTTPS) yang diekspos oleh domain. Untuk informasi selengkapnya, lihat the section called “Dukungan VPC”.
Peran IAM	Pertahankan peran default. OpenSearch Layanan menggunakan peran yang telah ditentukan ini (juga dikenal sebagai peran terkait layanan) untuk mengakses VPC Anda dan untuk menempatkan titik akhir VPC dan antarmuka jaringan di subnet VPC. Untuk informasi selengkapnya, lihat Peran yang terhubung dengan layanan untuk akses VPC.
Jenis Alamat IP	Pilih dual stack atau IPv4 sebagai jenis alamat IP Anda. Dual stack memungkinkan Anda untuk berbagi sumber daya domain di seluruh jenis alamat IPv4 dan IPv6, dan merupakan opsi yang disarankan. Jika Anda mengatur jenis alamat IP Anda ke tumpukan ganda, Anda tidak dapat mengubah jenis alamat Anda nanti.

20. Aktifkan atau nonaktifkan kontrol akses berbutir halus:

    - Jika Anda ingin menggunakan IAM untuk manajemen pengguna, pilih Setel IAM ARN sebagai pengguna utama dan tentukan ARN untuk peran IAM.
    - Jika Anda ingin menggunakan database pengguna internal, pilih Buat pengguna utama dan tentukan nama pengguna dan kata sandi.

    Opsi apa pun yang Anda pilih, pengguna master dapat mengakses semua indeks di cluster dan semua API. OpenSearch Untuk panduan pilihan mana yang harus dipilih, lihat the section called “Konsep utama”.

Jika Anda menonaktifkan kontrol akses detail, Anda masih dapat mengontrol akses ke domain Anda dengan menempatkannya dalam VPC, menerapkan kebijakan akses yang ketat, atau keduanya. Anda harus mengaktifkan node-to-node enkripsi dan enkripsi saat istirahat untuk menggunakan kontrol akses berbutir halus.

> **Note**
>
> Kami sangat menyarankan agar mengaktifkan kontrol akses detail untuk melindungi data di domain Anda. Kontrol akses detail menyediakan keamanan di tingkat klaster, indeks, dokumen, dan bidang.

21. (Opsional) Jika Anda ingin menggunakan otentikasi SAMB untuk OpenSearch Dasbor, pilih Aktifkan otentikasi SAMB dan konfigurasikan opsi SAFL untuk domain. Untuk petunjuk, lihat the section called “Otentikasi SAMP untuk Dasbor OpenSearch ”.
22. (Opsional) Jika Anda ingin menggunakan otentikasi Amazon Cognito untuk OpenSearch Dasbor, pilih Aktifkan otentikasi Amazon Cognito. Kemudian pilih kumpulan pengguna Amazon Cognito dan kumpulan identitas yang ingin Anda gunakan untuk otentikasi OpenSearch Dasbor. Untuk panduan tentang membuat sumber daya ini, lihat the section called “Otentikasi Amazon Cognito untuk Dasbor OpenSearch ”.
23. Untuk kebijakan Access, pilih kebijakan akses atau konfigurasikan kebijakan Anda sendiri. Jika Anda memilih untuk membuat kebijakan khusus, Anda dapat mengonfigurasinya sendiri atau mengimpornya dari domain lain. Untuk informasi selengkapnya, lihat the section called “Manajemen Identitas dan Akses”.

    > **Note**
    >
    > Jika Anda mengaktifkan akses VPC, Anda tidak dapat menggunakan kebijakan berbasis IP. Sebagai gantinya, Anda bisa menggunakan grup keamanan untuk mengontrol alamat IP yang dapat mengakses domain. Untuk informasi selengkapnya, lihat the section called “Tentang kebijakan akses pada domain VPC”.

24. (Opsional) Untuk mengharuskan semua permintaan ke domain tiba melalui HTTPS, pilih Memerlukan HTTPS untuk semua lalu lintas ke domain. Untuk mengaktifkan node-to-node enkripsi, pilih ode-to-nodeenkripsi N. Untuk informasi selengkapnya, lihat the section called “ode-to-node Enkripsi N”. Untuk mengaktifkan enkripsi data saat istirahat, pilih Aktifkan enkripsi data

saat istirahat. Opsi ini telah dipilih sebelumnya jika Anda memilih opsi penyebaran Multi-AZ dengan Siaga.

25. (Opsional) Pilih Gunakan kunci yang AWS dimiliki agar OpenSearch Layanan membuat kunci AWS KMS enkripsi atas nama Anda (atau gunakan kunci yang sudah dibuat). Jika tidak, pilih kunci KMS Anda sendiri. Untuk informasi selengkapnya, lihat the section called “Enkripsi diam”.
26. Untuk jendela Off-peak, pilih waktu mulai untuk menjadwalkan pembaruan perangkat lunak layanan dan pengoptimalan Auto-Tune yang memerlukan penerapan biru/hijau. Pembaruan off-peak membantu meminimalkan ketegangan pada node master khusus cluster selama periode lalu lintas tinggi.
27. Untuk Auto-Tune, pilih apakah akan mengizinkan OpenSearch Layanan menyarankan perubahan konfigurasi terkait memori ke domain Anda untuk meningkatkan kecepatan dan stabilitas. Untuk informasi selengkapnya, lihat the section called “Auto-Tune”.

    (Opsional) Pilih jendela Off-peak untuk menjadwalkan jendela berulang di mana Auto-Tune memperbarui domain.
28. (Opsional) Pilih Pembaruan perangkat lunak otomatis untuk mengaktifkan pembaruan perangkat lunak otomatis.
29. (Opsional) Tambahkan tag untuk mendeskripsikan domain Anda sehingga Anda dapat mengkategorikan dan memfilter informasi tersebut. Untuk informasi selengkapnya, lihat the section called “Penandaan domain”.
30. (Opsional) Perluas dan konfigurasikan pengaturan cluster lanjutan. Untuk rangkuman opsi ini, lihat the section called “Pengaturan cluster lanjutan”.
31. Pilih Buat.

Membuat domain OpenSearch Layanan ()AWS CLI

Alih-alih membuat domain OpenSearch Layanan dengan menggunakan konsol, Anda dapat menggunakan AWS CLI. Untuk sintaks, lihat Amazon OpenSearch Service dalam referensi perintah AWS CLI a.

Contoh perintah

Contoh pertama ini menunjukkan konfigurasi domain OpenSearch Layanan berikut:

- Membuat domain OpenSearch Layanan bernama mylogs dengan OpenSearch versi 1.2
- Mempopulasikan domain dengan dua instans dari tipe instans `r6g.large.search`

- Menggunakan volume gp3 EBS 100 GiB General Purpose (SSD) untuk penyimpanan untuk setiap node data
- Memungkinkan akses anonim, tetapi hanya dari satu alamat IP: 192.0.2.0/32

```
aws opensearch create-domain \
    --domain-name mylogs \
    --engine-version OpenSearch_1.2 \
    --cluster-config  InstanceType=r6g.large.search,InstanceCount=2 \
    --ebs-options
 EBSEnabled=true,VolumeType=gp3,VolumeSize=100,Iops=3500,Throughput=125 \
    --access-policies '{"Version": "2012-10-17", "Statement": [{"Action": "es:*",
 "Principal":"*","Effect": "Allow", "Condition": {"IpAddress":{"aws:SourceIp":
["192.0.2.0/32"]}}}]}'
```

Contoh berikutnya menunjukkan konfigurasi domain OpenSearch Layanan berikut:

- Membuat domain OpenSearch Layanan bernama mylogs dengan Elasticsearch versi 7.10
- Mempopulasikan domain dengan enam instans dari tipe instans `r6g.large.search`
- Menggunakan volume gp2 EBS 100 GiB General Purpose (SSD) untuk penyimpanan untuk setiap node data
- Membatasi akses ke layanan untuk satu pengguna, diidentifikasi oleh Akun AWS ID pengguna: 555555555555
- Mendistribusikan beberapa instans di seluruh Availability Zone

```
aws opensearch create-domain \
    --domain-name mylogs \
    --engine-version Elasticsearch_7.10 \
    --cluster-config
 InstanceType=r6g.large.search,InstanceCount=6,ZoneAwarenessEnabled=true,ZoneAwarenessConfig={A
 \
    --ebs-options EBSEnabled=true,VolumeType=gp2,VolumeSize=100 \
    --access-policies '{"Version": "2012-10-17", "Statement": [ { "Effect": "Allow",
 "Principal": {"AWS": "arn:aws:iam::555555555555:root" }, "Action":"es:*", "Resource":
 "arn:aws:es:us-east-1:555555555555:domain/mylogs/*" } ] }'
```

Contoh berikutnya menunjukkan konfigurasi domain OpenSearch Layanan berikut:

- Membuat domain OpenSearch Layanan bernama mylogs dengan OpenSearch versi 1.0

- Mempopulasikan domain dengan sepuluh instans dari tipe instans `r6g.xlarge.search`
- Mempopulasikan domain dengan sepuluh instans dari tipe instans `r6g.large.search` untuk melayani sebagai simpul utama khusus
- Menggunakan volume Provisioned IOPS EBS 100 GiB untuk penyimpanan, dikonfigurasi dengan performa dasar 1000 IOPS untuk setiap simpul data
- Membatasi akses ke satu pengguna dan ke satu subsumber daya, API `_search`

```
aws opensearch create-domain \
    --domain-name mylogs \
    --engine-version OpenSearch_1.0 \
    --cluster-config
 InstanceType=r6g.xlarge.search,InstanceCount=10,DedicatedMasterEnabled=true,DedicatedMasterTyp
 \
    --ebs-options EBSEnabled=true,VolumeType=io1,VolumeSize=100,Iops=1000 \
    --access-policies '{"Version": "2012-10-17", "Statement": [ { "Effect": "Allow",
 "Principal": { "AWS": "arn:aws:iam::555555555555:root" }, "Action": "es:*",
 "Resource": "arn:aws:es:us-east-1:555555555555:domain/mylogs/_search" } ] }'
```

> **Note**
>
> Jika Anda mencoba membuat domain OpenSearch Layanan dan domain dengan nama yang sama sudah ada, CLI tidak melaporkan kesalahan. Sebaliknya, CLI mengembalikan detail untuk domain yang ada.

Membuat domain OpenSearch Layanan (AWS SDK)

AWS SDK (kecuali SDK Android dan iOS) mendukung semua tindakan yang ditentukan dalam Referensi API OpenSearch Layanan Amazon, termasuk. `CreateDomain` Untuk kode sampel, lihat the section called “Menggunakan AWS SDKs”. Untuk informasi selengkapnya tentang menginstal dan menggunakan AWS SDK, lihat Kit Pengembangan AWS Perangkat Lunak.

Membuat domain OpenSearch Layanan ()AWS CloudFormation

OpenSearch Layanan terintegrasi dengan AWS CloudFormation, layanan yang membantu Anda memodelkan dan mengatur AWS sumber daya Anda sehingga Anda dapat menghabiskan lebih sedikit waktu untuk membuat dan mengelola sumber daya dan infrastruktur Anda. Anda

membuat template yang menjelaskan OpenSearch domain yang ingin Anda buat, dan menyediakan CloudFormation serta mengonfigurasi domain untuk Anda. Untuk informasi selengkapnya, termasuk contoh templat JSON dan YAMAL untuk OpenSearch domain, lihat referensi jenis sumber daya Amazon OpenSearch Service di Panduan Pengguna.AWS CloudFormation

Mengonfigurasi kebijakan akses

Amazon OpenSearch Service menawarkan beberapa cara untuk mengonfigurasi akses ke domain OpenSearch Layanan Anda. Untuk informasi lebih lanjut, lihat the section called “Manajemen Identitas dan Akses” dan the section called “Kontrol akses detail”.

Konsol menyediakan kebijakan akses yang telah dikonfigurasi yang dapat Anda sesuaikan untuk kebutuhan spesifik domain Anda. Anda juga dapat mengimpor kebijakan akses dari domain OpenSearch Layanan lain. Untuk informasi tentang bagaimana kebijakan akses ini berinteraksi dengan akses VPC, lihat the section called “Tentang kebijakan akses pada domain VPC”.

Untuk mengonfigurasi kebijakan akses (konsol)

1. Masuk ke https://aws.amazon.com, dan kemudian pilih Masuk ke Konsol.
2. Di bawah Analytics, pilih OpenSearch Layanan Amazon.
3. Di panel navigasi, di bawah Domain, pilih domain yang ingin Anda perbarui.
4. Pilih Tindakan dan Edit konfigurasi keamanan.
5. Edit kebijakan akses JSON, atau impor opsi yang telah dikonfigurasi sebelumnya.
6. Pilih Simpan perubahan.

Pengaturan cluster lanjutan

Gunakan opsi lanjutan untuk mengonfigurasi berikut ini:

Indeks di badan permintaan

Menentukan apakah referensi eksplisit untuk indeks diperbolehkan di dalam badan permintaan HTTP. Mengatur properti ini ke `false` mencegah pengguna melewati kontrol akses untuk sub sumber daya. Secara default, nilainya adalah `true`. Untuk informasi selengkapnya, lihat the section called “Pilihan lanjutan dan pertimbangan API”.

Alokasi cache Fielddata

Menentukan persentase ruang timbunan Java yang dialokasikan ke data bidang. Secara default, pengaturan ini adalah 20% dari timbunan JVM.

> **Note**
>
> Banyak pelanggan menanyakan indeks harian yang berputar. Kami menyarankan Anda untuk memulai pengujian benchmark dengan `indices.fielddata.cache.size` dikonfigurasi ke 40% dari timbunan JVM untuk sebagian besar kasus penggunaan ini. Untuk indeks yang sangat besar, Anda mungkin memerlukan cache data bidang yang besar.

Jumlah klausa maks

Menentukan jumlah maksimum klausul yang diperbolehkan dalam kueri boolean Lucene. Secara default adalah 1.024. Kueri dengan lebih dari jumlah klausa yang diizinkan menghasilkan kesalahan `TooManyClauses`. Untuk informasi selengkapnya, lihat dokumentasi Lucene.

Membuat perubahan konfigurasi di Amazon OpenSearch Service

Amazon OpenSearch Service menggunakan proses penyebaran biru/hijau saat memperbarui domain. Penerapan biru/hijau menciptakan lingkungan idle untuk pembaruan domain yang menyalin lingkungan produksi, dan mengarahkan pengguna ke lingkungan baru setelah pembaruan tersebut selesai. Dalam deployment blue/green, lingkungan biru adalah lingkungan produksi saat ini. Lingkungan hijau adalah lingkungan yang menganggur.

Data dimigrasikan dari lingkungan biru ke lingkungan hijau. Ketika lingkungan baru siap, OpenSearch Layanan beralih ke lingkungan untuk mempromosikan lingkungan hijau menjadi lingkungan produksi baru. Peralihan terjadi tanpa kehilangan data. Praktik ini meminimalkan waktu henti dan mempertahankan lingkungan asli jika penerapan ke lingkungan baru tidak berhasil.

Topik

- Memulai dan melacak perubahan konfigurasi
- Tahapan perubahan konfigurasi
- Biaya untuk perubahan konfigurasi
- Memecahkan masalah kesalahan validasi

Perubahan yang biasanya menyebabkan penerapan biru/hijau

Operasi berikut menyebabkan deployment biru/hijau:

- Mengubah tipe instans
- Mengaktifkan kontrol akses detail
- Melakukan pembaruan perangkat lunak layanan
- Jika domain Anda tidak memiliki simpul utama yang berdedikasi, mengubah jumlah instans data
- Mengaktifkan atau menonaktifkan simpul utama khusus
- Mengaktifkan atau menonaktifkan Multi-AZ tanpa Siaga
- Mengubah jenis penyimpanan, jenis volume, atau ukuran volume
- Memilih subnet VPC yang berbeda
- Menambahkan atau menghapus grup keamanan VPC
- Mengaktifkan atau menonaktifkan otentikasi Amazon Cognito untuk Dasbor OpenSearch
- Memilih kolam pengguna atau kolam identitas Amazon Cognito yang berbeda
- Mengubah pengaturan lanjutan
- Memutakhirkan ke OpenSearch versi baru (OpenSearch Dasbor mungkin tidak tersedia selama beberapa atau semua peningkatan)
- Mengaktifkan enkripsi data saat istirahat atau node-to-node enkripsi
- Mengaktifkan atau menonaktifkan UltraWarm atau penyimpanan dingin
- Menonaktifkan Auto-Tune dan memutar kembali perubahannya
- Mengaitkan plugin opsional ke domain dan memisahkan plugin opsional dari domain
- Meningkatkan jumlah node master khusus untuk domain dengan dua node master khusus dan kesadaran zona diaktifkan
- Mengurangi ukuran volume EBS
- Mengubah ukuran volume EBS, IOPS, atau throughput, jika perubahan terakhir yang Anda buat sedang berlangsung atau terjadi kurang dari 6 jam yang lalu

- Mengaktifkan publikasi log audit ke CloudWatch.

Untuk Multi-AZ dengan domain Siaga, Anda hanya dapat membuat satu permintaan perubahan pada satu waktu. Jika perubahan sedang berlangsung, permintaan baru akan ditolak. Anda dapat memeriksa status perubahan saat ini dengan `DescribeDomainChangeProgress` API.

Perubahan yang biasanya tidak menyebabkan penerapan biru/hijau

Dalam banyak kasus, operasi berikut tidak menyebabkan deployment biru/hijau:

- Mengubah kebijakan akses
- Memodifikasi titik akhir kustom
- Mengubah kebijakan Transport Layer Security (TLS)
- Mengubah jam snapshot otomatis
- Mengaktifkan atau menonaktifkan Memerlukan HTTPS
- Mengaktifkan Auto-Tune atau menonaktifkannya tanpa memutar kembali perubahannya
- Jika domain Anda memiliki node master khusus atau mengubah jumlah UltraWarm node
- Mengubah jumlah node data
- Jika domain Anda memiliki node master khusus, ubah jenis instans master khusus atau jumlah node (kecuali untuk domain dengan dua master khusus dan kesadaran zona diaktifkan)
- Mengaktifkan atau menonaktifkan publikasi log kesalahan atau memperlambat log ke CloudWatch
- Menonaktifkan publikasi log audit ke. CloudWatch
- Meningkatkan ukuran volume, mengubah tipe volume, IOPS, dan throughput hingga 3 TiB per ukuran volume node data
- Menambahkan atau menghapus tag

> **Note**
>
> Ada beberapa pengecualian tergantung pada versi perangkat lunak layanan Anda. Jika Anda ingin benar-benar yakin bahwa perubahan tidak akan menyebabkan penyebaran biru/hijau, lakukan dry run sebelum memperbarui domain Anda, jika opsi ini tersedia. Beberapa perubahan tidak menawarkan opsi dry run. Kami biasanya menyarankan agar Anda membuat perubahan pada cluster Anda di luar jam lalu lintas puncak.

Menentukan apakah perubahan akan menyebabkan penyebaran biru/hijau

Anda dapat menguji beberapa jenis perubahan konfigurasi yang direncanakan untuk menentukan apakah perubahan tersebut akan menyebabkan penerapan biru/hijau, tanpa harus berkomitmen pada perubahan tersebut. Sebelum Anda memulai perubahan konfigurasi, gunakan konsol atau API untuk menjalankan pemeriksaan validasi guna memastikan bahwa domain Anda memenuhi syarat untuk pembaruan.

Console

Untuk memvalidasi perubahan konfigurasi

1. Arahkan ke konsol OpenSearch Layanan Amazon dihttps://console.aws.amazon.com/aos/.
2. Di panel navigasi kiri, pilih Domain.
3. Pilih domain yang ingin Anda ubah konfigurasi. Ini membuka halaman detail domain. Pilih menu tarik-turun Tindakan dan kemudian pilih Edit konfigurasi cluster.
4. Pada halaman konfigurasi cluster Edit, Anda dapat membuat perubahan pada jenis instance, jumlah node, dan konfigurasi lainnya. Setelah mengonfirmasi perubahan di panel ringkasan, pilih Jalankan.
5. Setelah dry run Anda selesai, hasilnya secara otomatis ditampilkan di bagian bawah halaman, bersama dengan ID dry run. Hasil ini memberi tahu Anda kategori mana yang termasuk dalam perubahan Anda:

   - Memulai penerapan biru/hijau
   - Tidak memerlukan penerapan biru/hijau
   - Berisi kesalahan validasi yang perlu Anda atasi sebelum Anda dapat menyimpan perubahan Anda

   Perhatikan bahwa setiap dry run menimpa yang sebelumnya. Untuk mencari detail setiap dry run nanti, pastikan Anda menyimpan ID dry run Anda. Setiap dry run tersedia selama 90 hari, atau sampai Anda membuat pembaruan konfigurasi.
6. Untuk melanjutkan pembaruan konfigurasi Anda, pilih Simpan perubahan. Jika tidak, pilih Batalkan. Opsi mana pun membawa Anda kembali ke tab konfigurasi Cluster. Pada tab ini, Anda dapat memilih Detail Dry run untuk melihat detail dry run terbaru Anda. Halaman ini juga mencakup side-by-side perbandingan antara konfigurasi sebelum dry run dan konfigurasi dry run.

API

Anda dapat melakukan validasi dry run melalui API konfigurasi. Untuk menguji perubahan Anda dengan API, setel `DryRun` ke`true`, dan `DryRunMode` ke`Verbose`. Mode verbose menjalankan pemeriksaan validasi selain menentukan apakah perubahan akan memulai penerapan biru/hijau. Misalnya, UpdateDomainConfigpermintaan ini menguji jenis penerapan yang dihasilkan dari pengaktifan UltraWarm:

```
POST https://es.us-east-1.amazonaws.com/2021-01-01/opensearch/domain/my-domain/
config
{
   "ClusterConfig": {
    "WarmCount": 3,
    "WarmEnabled": true,
    "WarmType": "ultrawarm1.large.search"
   },
   "DryRun": true,
   "DryRunMode": "Verbose"
}
```

Permintaan menjalankan pemeriksaan validasi dan mengembalikan jenis penerapan yang akan menyebabkan perubahan tetapi tidak benar-benar melakukan pembaruan:

```
{
   "ClusterConfig": {
     ...
    },
   "DryRunResults": {
      "DeploymentType": "Blue/Green",
      "Message": "This change will require a blue/green deployment."
    }
}
```

Jenis penerapan yang mungkin adalah:

- `Blue/Green`— Perubahan akan menyebabkan penyebaran biru/hijau.
- `DynamicUpdate`— Perubahan tidak akan menyebabkan penyebaran biru/hijau.
- `Undetermined`— Domain masih dalam status pemrosesan, sehingga jenis penerapan tidak dapat ditentukan.
- `None`- Tidak ada perubahan konfigurasi.

Jika validasi gagal, ia mengembalikan daftar kegagalan validasi.

```
{
   "ClusterConfig":{
      "..."
   },
   "DryRunProgressStatus":{
      "CreationDate":"2023-01-12T01:14:33.847Z",
      "DryRunId":"db00ca39-48b2-4774-bbd3-252cf094d205",
      "DryRunStatus":"failed",
      "UpdateDate":"2023-01-12T01:14:33.847Z",
      "ValidationFailures":[
         {
            "Code":"Cluster.Index.WriteBlock",
            "Message":"Cluster has index write blocks."
         }
      ]
   }
}
```

Jika statusnya diampending, Anda dapat menggunakan ID dry run dalam UpdateDomainConfig respons Anda dalam DescribeDryRunProgresspanggilan berikutnya untuk memeriksa status validasi.

```
GET https://es.us-east-1.amazonaws.com/2021-01-01/opensearch/domain/my-domain/
dryRun?dryRunId=my-dry-run-id
{
    "DryRunConfig": null,
    "DryRunProgressStatus": {
        "CreationDate": "2023-01-12T01:14:42.998Z",
        "DryRunId": "db00ca39-48b2-4774-bbd3-252cf094d205",
        "DryRunStatus": "succeeded",
        "UpdateDate": "2023-01-12T01:14:49.334Z",
        "ValidationFailures": null
    },
    "DryRunResults": {
        "DeploymentType": "Blue/Green",
        "Message": "This change will require a blue/green deployment."
    }
}
```

Untuk menjalankan analisis dry run tanpa pemeriksaan validasi, setel `DryRunMode` ke `Basic` saat Anda menggunakan API konfigurasi.

Python

Kode Python berikut menggunakan UpdateDomainConfigAPI untuk melakukan pemeriksaan validasi dry run dan, jika pemeriksaan berhasil, memanggil API yang sama tanpa dry run untuk memulai pembaruan. Jika pemeriksaan gagal, skrip mencetak kesalahan dan berhenti.

```
import time
import boto3

client = boto3.client('opensearch')

response = client.UpdateDomainConfig(
    ClusterConfig={
        'WarmCount': 3,
        'WarmEnabled': True,
        'WarmCount': 123,
    },
    DomainName='test-domain',
    DryRun=True,
    DryRunMode='Verbose'
)

dry_run_id = response.DryRunProgressStatus.DryRunId

retry_count = 0

while True:

    if retry_count == 5:
        print('An error occured')
        break

    dry_run_progress_response = client.DescribeDryRunProgress('test-domain',
 dry_run_id)
    dry_run_status = dry_run_progress_response.DryRunProgressStatus.DryRunStatus

    if dry_run_status == 'succeeded':
        client.UpdateDomainConfig(
            ClusterConfig={
            'WarmCount': 3,
            'WarmEnabled': True,
```

```
            'WarmCount': 123,
        })
        break

    elif dry_run_status == 'failed':
        validation_failures_list =
dry_run_progress_response.DryRunProgressStatus.ValidationFailures
        for item in validation_failures_list:
            print(f"Code: {item['Code']}, Message: {item['Message']}")
        break

    retry_count += 1
    time.sleep(30)
```

Memulai dan melacak perubahan konfigurasi

> **Note**
>
> Anda dapat meminta satu perubahan konfigurasi pada satu waktu. Anda juga dapat mengelompokkan beberapa perubahan konfigurasi dalam satu permintaan. Tunggu status domain Anda menjadi `Active` sebelum meminta perubahan konfigurasi tambahan.

Anda dapat melihat bidang Status Pemrosesan Domain dan Ubah Status Konfigurasi di konsol OpenSearch Layanan Amazon untuk melacak perubahan domain dan konfigurasi. Anda juga dapat melacak perubahan domain dan konfigurasi melalui `ConfigChangeStatus` parameter `DomainProcessingStatus` dan dalam respons API. Untuk informasi selengkapnya, lihat tipe DomainStatusdata dalam referensi API OpenSearch Layanan.

Visibilitas status pemrosesan domain: Anda dapat dengan mudah menentukan status konfigurasi domain dengan melihat bidang Status Pemrosesan Domain di konsol. Demikian pula, parameter `DomainProcessingStatus` API dapat digunakan untuk mengidentifikasi status. Nilai-nilai berikut memproses status untuk domain:

- `Active`: Tidak ada perubahan konfigurasi yang sedang berlangsung. Anda dapat mengirimkan permintaan perubahan konfigurasi baru.
- `Creating`: Domain sedang dibuat.

- `Modifying`: Perubahan konfigurasi, seperti penambahan node data baru, EBS, gp3, penyediaan IOPS, atau pengaturan kunci KMS, sedang berlangsung.

  > **Note**
  >
  > Anda mungkin melihat status seperti `Modifying` dalam situasi di mana domain memerlukan gerakan pecahan untuk menyelesaikan perubahan konfigurasi. Untuk kompatibilitas mundur, perilaku `Processing` parameter tetap tidak berubah dalam respons API, dan disetel ke false segera setelah perubahan konfigurasi inti selesai, tanpa menunggu penyelesaian gerakan pecahan.

- `Upgrading Engine Version`: Upgrade versi mesin sedang berlangsung.
- `Updating Service Software`: Pembaruan perangkat lunak layanan sedang berlangsung.
- `Deleting`: Domain sedang dihapus.
- `Isolated`: Domain ditangguhkan.

Visibilitas status konfigurasi: Perubahan konfigurasi dapat dimulai oleh operator (misalnya penambahan node data baru, perubahan tipe instance) atau oleh layanan (misalnya Auto-Tune dan pembaruan off-peak hour). Anda dapat menemukan status detail perubahan konfigurasi terbaru di bidang Status Perubahan Konfigurasi pada konsol OpenSearch Layanan Amazon, dan di respons `ConfigChangeStatus` API. Nilai berikut menunjukkan status konfigurasi domain:

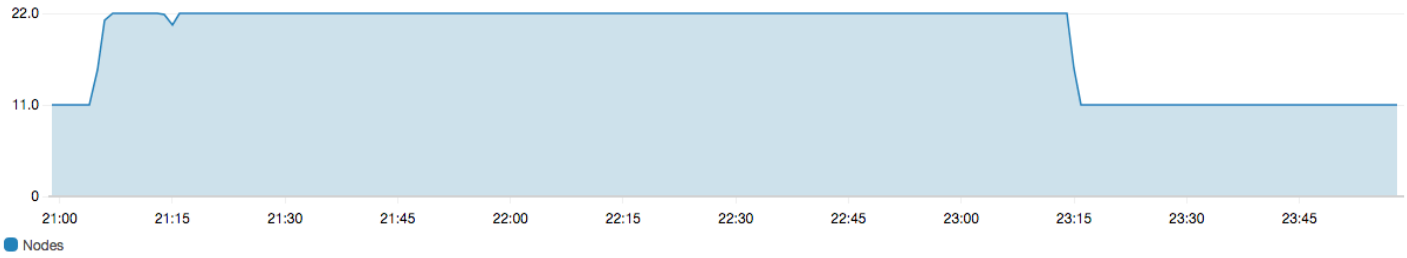
- `Pending`: Permintaan perubahan konfigurasi telah dikirimkan.
- `Initializing`: Layanan menginisialisasi permintaan perubahan konfigurasi.
- `Validating`: Layanan memvalidasi perubahan yang diminta dan sumber daya yang diperlukan.
- `Awaiting user inputs`: Berlaku ketika operator mengharapkan beberapa perubahan konfigurasi seperti perubahan jenis instance untuk melanjutkan lebih jauh. Anda dapat mengedit perubahan konfigurasi.
- `Applying changes`: Layanan menerapkan perubahan konfigurasi yang diminta.
- `Cancelled`: Perubahan konfigurasi dibatalkan. Jika Anda menerima status gagal validasi, Anda dapat mengklik Batal di konsol atau memanggil operasi `CancelDomainConfigChange` API. Jika Anda melakukan ini, semua perubahan yang diterapkan digulung kembali.
- `Completed`: Perubahan konfigurasi yang diminta telah selesai dengan sukses.
- `Validation Failed`: Perubahan yang diminta gagal validasi. Tidak ada perubahan konfigurasi yang diterapkan.

> **Note**
>
> Kegagalan validasi dapat disebabkan oleh indeks merah yang ada di domain Anda, tidak tersedianya jenis instans yang dipilih, atau ruang disk yang rendah. Untuk daftar kesalahan validasi, lihatthe section called “Memecahkan masalah kesalahan validasi”. Selama peristiwa kegagalan validasi, Anda dapat membatalkan, mencoba lagi, atau mengedit perubahan konfigurasi.

Ringkasan API: Anda dapat menggunakan operasi `DescribeDomainDescribeDomainChangeProgress`,, dan `DescribeDomainConfig` API untuk mendapatkan status pembaruan konfigurasi terperinci. Selain itu, Anda dapat menggunakan `CancelDomainConfigChange` untuk membatalkan pembaruan jika terjadi kegagalan validasi. Untuk informasi selengkapnya, lihat dokumentasi API OpenSearch Layanan

Ketika perubahan konfigurasi selesai, status domain berubah kembali ke. `Active`

Anda dapat meninjau kesehatan klaster dan CloudWatch metrik Amazon dan melihat bahwa jumlah node di cluster untuk sementara meningkat—seringkali berlipat ganda—saat pembaruan domain terjadi. Dalam ilustrasi berikut, Anda dapat melihat jumlah node berlipat ganda dari 11 menjadi 22 selama perubahan konfigurasi dan kembali ke 11 saat pembaruan selesai.



Peningkatan sementara ini dapat membebani simpul utama khusus klaster, yang tiba-tiba mungkin memiliki lebih banyak simpul untuk dikelola. Ini juga dapat meningkatkan latensi pencarian dan pengindeksan karena OpenSearch Layanan menyalin data dari cluster lama ke yang baru. Penting untuk mempertahankan kapasitas yang cukup di cluster untuk menangani overhead yang terkait dengan penerapan biru/hijau ini.

> **Important**
>
> Anda tidak dikenakan biaya tambahan selama perubahan konfigurasi dan pemeliharaan layanan. Anda ditagih hanya untuk jumlah node yang Anda minta untuk klaster Anda. Untuk spesifik, lihat the section called “Biaya untuk perubahan konfigurasi”.

Untuk mencegah kelebihan beban node master khusus, Anda dapat memantau penggunaan dengan CloudWatch metrik Amazon. Untuk nilai maksimum yang disarankan, lihat the section called “ CloudWatch Alarm yang direkomendasikan”.

Tahapan perubahan konfigurasi

Setelah Anda memulai perubahan konfigurasi, OpenSearch Layanan akan melalui serangkaian langkah untuk memperbarui domain Anda. Anda dapat melihat kemajuan perubahan konfigurasi di bawah Status perubahan konfigurasi di konsol. Langkah-langkah tepat yang dilalui pembaruan tergantung pada jenis perubahan yang Anda buat. Anda juga dapat memantau perubahan konfigurasi menggunakan operasi DescribeDomainChangeProgressAPI.

Berikut ini adalah tahapan yang mungkin dilakukan pembaruan selama perubahan konfigurasi:

Nama panggung	Deskripsi
Validasi	Memvalidasi bahwa domain memenuhi syarat untuk pembaruan , dan memunculkan masalah validasi jika perlu.
Menciptakan lingkungan baru	Menyelesaikan prasyarat yang diperlukan dan membuat

Nama panggung	Deskripsi
	sumber daya yang diperlukan untuk memulai penerapan biru/hijau.
Penyediaan node baru	Membuat serangkaian instance baru di lingkungan baru.
Perutean lalu lintas pada node baru	Mengarahkan lalu lintas ke node data yang baru dibuat.
Perutean lalu lintas pada node lama	Menonaktifkan lalu lintas pada node data lama.

Nama panggung	Deskripsi
Mempersiapkan node untuk dihapus	Bersiap untuk menghapus node. Langkah ini hanya terjadi ketika Anda menurunkan skala domain Anda (misalnya, dari 8 node menjadi 6 node).
Menyalin pecahan ke node baru	Memindahkan pecahan dari node lama ke node baru.
Mengakhiri node	Mengakhiri dan menghapus node lama setelah pecahan dihapus.

Nama panggung	Deskripsi
Menghapus sumber daya yang lebih lama	Menghapus sumber daya yang terkait dengan lingkunga n lama (misalnya penyeimbang beban).
Pembaruan dinamis	Ditampilk an ketika pembaruan tidak memerluka n penerapan biru/hijau dan dapat diterapka n secara dinamis.
Menerapkan perubahan terkait master khusus	Ditampilk an ketika jenis atau jumlah instans master khusus diubah.
Menerapkan perubahan terkait volume	Ditampilkan ketika ukuran volume, jenis, IOPS dan throughput diubah.

Biaya untuk perubahan konfigurasi

Jika Anda mengubah konfigurasi untuk domain, OpenSearch Service akan membuat klaster baru seperti yang dijelaskan dalamthe section called “Perubahan konfigurasi”. Selama migrasi dari lama ke baru, Anda dikenakan biaya berikut:

- Jika Anda mengubah tipe instans, Anda akan dikenakan biaya untuk kedua klaster selama satu jam pertama. Setelah satu jam pertama, Anda hanya dikenakan biaya untuk klaster baru. Volume EBS tidak dikenakan biaya dua kali karena merupakan bagian dari klaster Anda, sehingga penagihannya mengikuti penagihan instans.

  Contoh: Anda mengubah konfigurasi dari tiga instans `m3.xlarge` ke empat instans `m4.large`. Untuk jam pertama, Anda dikenakan biaya untuk kedua cluster (3 * `m3.xlarge` + 4 *`m4.large`). Setelah satu jam pertama, Anda hanya dikenakan biaya untuk cluster baru (4*`m4.large`).
- Jika Anda tidak mengubah jenis instans, Anda hanya dikenakan biaya untuk klaster terbesar selama satu jam pertama. Setelah satu jam pertama, Anda hanya dikenakan biaya untuk cluster baru.

  Contoh: Anda mengubah konfigurasi dari tiga enam instans `m3.xlarge` ke tiga instans `m3.xlarge`. Untuk jam pertama, Anda dikenakan biaya untuk cluster terbesar (6*`m3.xlarge`). Setelah satu jam pertama, Anda hanya dikenakan biaya untuk cluster baru (3*`m3.xlarge`).

Memecahkan masalah kesalahan validasi

Saat Anda memulai perubahan konfigurasi atau melakukan pemutakhiran versi OpenSearch atau Elasticsearch, OpenSearch Layanan terlebih dahulu melakukan serangkaian pemeriksaan validasi untuk memastikan bahwa domain Anda memenuhi syarat untuk pembaruan. Jika salah satu pemeriksaan ini gagal, Anda menerima pemberitahuan di konsol yang berisi masalah spesifik yang harus Anda perbaiki sebelum memperbarui domain Anda. Tabel berikut mencantumkan kemungkinan masalah domain yang mungkin muncul oleh OpenSearch Layanan, dan langkah-langkah untuk mengatasinya.

Isu	Kode kesalahan	Langkah pemecahan masalah
Grup keamanan tidak ditemukan	`SecurityGroupNotFound`	Grup keamanan yang terkait dengan domain OpenSearch Layanan Anda tidak ada. Untuk mengatasi masalah ini, buat grup keamanan dengan nama yang ditentukan.
Subnet tidak ditemukan	`SubnetNotFound`	Subnet yang terkait dengan domain OpenSearch Layanan Anda tidak ada. Untuk mengatasi masalah ini, buat subnet di VPC Anda.
Peran terkait layanan tidak dikonfigurasi	`SLRNotConfigured`	Peran terkait layanan untuk OpenSearch Layanan tidak dikonfigurasi. Peran terkait layanan telah ditentukan sebelumnya oleh OpenSearch Layanan dan mencakup semua izin yang diperlukan layanan untuk memanggil AWS layanan lain atas nama Anda. Jika peran tidak ada, Anda mungkin perlu membuatnya secara manual.
Alamat IP tidak cukup	`InsufficientFreeIPsForSubnets`	Satu atau lebih subnet VPC Anda tidak memiliki cukup alamat IP untuk memperbarui domain Anda. Untuk menghitung berapa banyak alamat IP yang Anda butuhkan, lihatthe section called “Menyimpan alamat IP di subnet VPC”.
Kumpulan pengguna Cognito tidak ada	`CognitoUserPoolNotFound`	OpenSearch Layanan tidak dapat menemukan kumpulan pengguna Amazon Cognito. Konfirmasi bahwa Anda membuat satu dan memiliki ID yang benar. Untuk menemukan ID, Anda dapat menggunakan konsol Amazon Cognito atau perintah AWS CLI berikut: `aws cognito-idp list-user-pools --max-results 60 --region us-east-1`
Kumpulan identitas Cognito tidak ada	`CognitoIdentityPoolNotFound`	OpenSearch Layanan tidak dapat menemukan kumpulan identitas Cognito. Konfirmasi bahwa Anda membuat satu dan memiliki ID yang benar. Untuk menemukan ID, Anda dapat menggunakan konsol Amazon Cognito atau perintah AWS CLI berikut:

Isu	Kode kesalahan	Langkah pemecahan masalah
		`aws cognito-identity list-identity-pools --max-results 60 --region us-east-1`
Domain Cognito tidak ditemukan untuk kumpulan pengguna	`CognitoDomainNotFound`	Kolam pengguna tidak memiliki nama domain. Anda dapat mengonfigurasinya menggunakan konsol Amazon Cognito atau perintah berikut: AWS CLI `aws cognito-idp create-user-pool-domain --domain my-domain --user-pool-id id`
Peran Cognito tidak dikonfigurasi	`CognitoRoleNotConfigured`	Peran IAM yang memberikan izin OpenSearch Layanan untuk mengonfigurasi pengguna Amazon Cognito dan kumpulan identitas, dan menggunakannya untuk otentikasi, tidak dikonfigurasi. Konfigurasikan peran dengan set izin dan hubungan kepercayaan yang sesuai. Anda dapat menggunakan konsol, yang membuat CognitoAccessForAmazonOpenSearchperan default untuk Anda, atau Anda dapat mengonfigurasi peran secara manual menggunakan AWS CLI atau AWS SDK.
Tidak dapat mendeskripsikan kumpulan pengguna	`UserPoolNotDescribable`	Peran Amazon Cognito yang ditentukan tidak memiliki izin untuk menjelaskan kumpulan pengguna yang terkait dengan domain Anda. Pastikan kebijakan izin peran memungkinkan `cognito-identity:DescribeUserPool` tindakan. Lihat the section called "Tentang peran CognitoAccessForAmazonOpenSearch" kebijakan izin lengkap.
Tidak dapat mendeskripsikan kumpulan identitas	`IdentityPoolNotDescribable`	Peran Amazon Cognito yang ditentukan tidak memiliki izin untuk menjelaskan kumpulan identitas yang terkait dengan domain Anda. Pastikan kebijakan izin peran memungkinkan `cognito-identity:DescribeIdentityPool` tindakan. Lihat the section called "Tentang peran CognitoAccessForAmazonOpenSearch" kebijakan izin lengkap.

Isu	Kode kesalahan	Langkah pemecahan masalah
Tidak dapat mendeskripsikan pengguna dan kumpulan identitas	`CognitoPoolsNotDescribable`	Peran Amazon Cognito yang ditentukan tidak memiliki izin untuk mendeskripsikan kumpulan pengguna dan identitas yang terkait dengan domain Anda. Pastikan kebijakan izin peran memungkinkan `cognito-identity:DescribeIdentityPool` dan `cognito-identity:DescribeUserPool` tindakan. Lihat the section called “Tentang peran CognitoAccessForAmazonOpenSearch” kebijakan izin lengkap.
Kunci KMS tidak diaktifkan	`KMSKeyNotEnabled`	Kunci AWS Key Management Service (AWS KMS) yang digunakan untuk mengenkripsi domain Anda dinonaktifkan. Aktifkan kembali kunci segera.
Sertifikat kustom tidak dalam keadaan ISSUED	`InvalidCertificate`	Jika domain Anda menggunakan titik akhir kustom, Anda mengamankannya dengan membuat sertifikat SSL di AWS Certificate Manager (ACM) atau mengimpor salah satu dari Anda sendiri. Status sertifikat harus Diterbitkan. Jika Anda menerima kesalahan ini, periksa status sertifikat Anda di konsol ACM. Jika status validasi Kedaluwarsa, Gagal, Tidak Aktif, atau Tertunda, lihat dokumentasi pemecahan masalah ACM untuk menyelesaikan masalah.
Tidak cukup kapasitas untuk meluncurkan jenis instans yang dipilih	`InsufficientInstanceCapacity`	Kapasitas tipe instans yang diminta tidak tersedia. Misalnya, Anda mungkin telah meminta lima `i3.16xlarge.search` node, tetapi OpenSearch Layanan tidak memiliki cukup `i3.16xlarge.search` host yang tersedia, sehingga permintaan tidak dapat dipenuhi. Periksa jenis instans yang didukung di OpenSearch Layanan dan pilih jenis instans yang berbeda.
Indeks merah dalam cluster	`RedCluster`	Satu atau lebih indeks di cluster Anda memiliki status merah, yang mengarah ke status cluster merah secara keseluruhan. Untuk memecahkan masalah dan memperbaiki masalah ini, lihat. the section called “Status klaster merah”

Isu	Kode kesalahan	Langkah pemecahan masalah
Pemutus sirkuit memori, terlalu banyak permintaan	`TooManyRequests`	Ada terlalu banyak permintaan pencarian dan penulisan ke domain Anda, sehingga OpenSearch Layanan tidak dapat memperbarui konfigurasinya. Anda dapat mengurangi jumlah permintaan, menskalakan instans secara vertikal hingga 64 GiB RAM, atau menskalakan secara horizontal dengan menambahkan instance.
Konfigurasi baru tidak dapat menyimpan data (ruang disk rendah)	`InsufficientStorageCapacity`	Ukuran penyimpanan yang dikonfigurasi tidak dapat menampung semua data di domain Anda. Untuk mengatasi masalah ini, pilih volume yang lebih besar, hapus indeks yang tidak digunakan, atau tingkatkan jumlah node di cluster untuk segera membebaskan ruang disk.

Pecahan disematkan ke node tertentu	`ShardMovementBlocked`	Satu atau lebih indeks di domain Anda dilampirkan ke node tertentu dan tidak dapat dipindahkan. Ini kemungkinan besar terjadi karena Anda mengonfigurasi pemfilteran alokasi pecahan, yang memungkinkan Anda menentukan node mana yang diizinkan untuk meng-host pecahan indeks tertentu.
		
		Untuk mengatasi masalah ini, hapus filter alokasi pecahan dari semua indeks yang terpengaruh:
		
		`PUT my-index/_settings`
		`{`
		`  "settings": {`
		`    "index.routing.allocation.require._name": null`
		`  }`
		`}`

Isu	Kode kesalahan	Langkah pemecahan masalah
Konfigurasi baru tidak dapat menampung semua pecahan (jumlah pecahan)	`TooManyShards`	Jumlah pecahan pada domain Anda terlalu tinggi, yang mencegah OpenSearch Service memindahkannya ke konfigurasi baru. Untuk mengatasi masalah ini, skala domain Anda secara horizontal dengan menambahkan node dari jenis konfigurasi yang sama dengan node cluster Anda saat ini. Perhatikan bahwa ukuran volume EBS maksimum bergantung pada jenis instance node. Untuk mencegah masalah ini di masa mendatang, lihat the section called “Memilih jumlah serpihan” dan tentukan strategi sharding yang sesuai untuk kasus penggunaan Anda.
Subnet yang terkait dengan domain Anda tidak mendukung alamat IPv4	`ResultCodeIPv4BlockNotExists`	Untuk mengatasi masalah ini, buat subnet atau perbarui subnet yang ada di VPC Anda sesuai dengan jenis alamat IP domain yang dikonfigurasi. Jika domain Anda hanya menggunakan jenis alamat IPv4, gunakan subnet khusus IPv4. Jika domain Anda menggunakan mode Dual-stack, gunakan subnet dual-stack.
Subnet yang terkait dengan domain Anda tidak mendukung alamat IPv6	`ResultCodeIPv6BlockNotExists`	Untuk mengatasi masalah ini, buat subnet atau perbarui subnet yang ada di VPC Anda sesuai dengan jenis alamat IP domain yang dikonfigurasi. Jika domain Anda hanya menggunakan jenis alamat IPv4, gunakan subnet khusus IPv4. Jika domain Anda menggunakan mode Dual-stack, gunakan subnet dual-stack.

Pembaruan perangkat lunak layanan di Amazon OpenSearch Service

> **Note**
>
> Untuk penjelasan tentang perubahan dan penambahan yang dibuat di setiap pembaruan perangkat lunak layanan utama (non-patch), lihat catatan rilis.

Amazon OpenSearch Service secara teratur merilis pembaruan perangkat lunak layanan yang menambahkan fitur atau meningkatkan domain Anda. Panel Notifikasi di konsol adalah cara termudah untuk melihat apakah pembaruan tersedia atau untuk memeriksa status pembaruan. Setiap notifikasi menyertakan detail tentang pembaruan perangkat lunak layanan. Semua pembaruan perangkat lunak layanan menggunakan penerapan biru/hijau untuk meminimalkan waktu henti.

Pembaruan perangkat lunak layanan berbeda dari peningkatan OpenSearch versi. Untuk informasi tentang memutakhirkan ke versi yang lebih baru OpenSearch, lihatthe section called “Memutakhirkan domain”.

Topik

Pembaruan opsional versus yang diperlukan

OpenSearch Layanan memiliki dua kategori pembaruan perangkat lunak layanan yang luas:

Pembaruan opsional

Pembaruan perangkat lunak layanan opsional umumnya mencakup peningkatan dan dukungan untuk fitur atau fungsionalitas baru. Pembaruan opsional tidak diberlakukan pada domain Anda, dan

tidak ada tenggat waktu yang sulit untuk menginstalnya. Ketersediaan pembaruan dikomunikasikan melalui email dan pemberitahuan konsol. Anda dapat memilih untuk segera menerapkan pembaruan atau menjadwal ulang untuk tanggal dan waktu yang lebih tepat. Anda juga dapat menjadwalkannya selama jendela off-peak domain. Sebagian besar pembaruan perangkat lunak bersifat opsional.

Terlepas dari apakah Anda menjadwalkan pembaruan atau tidak, jika Anda membuat perubahan pada domain yang menyebabkan penyebaran biru/hijau, OpenSearch Layanan secara otomatis memperbarui perangkat lunak layanan Anda untuk Anda.

Anda dapat mengonfigurasi domain Anda untuk menerapkan pembaruan opsional secara otomatis selama jam sibuk. Ketika opsi ini diaktifkan, OpenSearch Layanan menunggu setidaknya 13 hari sejak pembaruan opsional tersedia dan kemudian menjadwalkan pembaruan setelah 72 jam (tiga hari). Anda menerima pemberitahuan konsol saat pembaruan dijadwalkan dan Anda dapat memilih untuk menjadwal ulang untuk kemudian hari.

Untuk mengaktifkan pembaruan perangkat lunak otomatis, pilih Aktifkan pembaruan perangkat lunak otomatis saat Anda membuat atau memperbarui domain Anda. Untuk mengonfigurasi pengaturan yang sama menggunakan AWS CLI, atur `--software-update-options` ke `true` saat Anda membuat atau memperbarui domain Anda.

Pembaruan yang diperlukan

Pembaruan perangkat lunak layanan yang diperlukan umumnya mencakup perbaikan keamanan penting atau pembaruan wajib lainnya untuk memastikan integritas dan fungsionalitas domain Anda yang berkelanjutan. Contoh pembaruan yang diperlukan adalah Log4j Common Vulnerabilities and Exposures (CVE) dan penegakan Instance Metadata Service Version 2 (IMDSv2). Jumlah pembaruan wajib dalam setahun biasanya kurang dari tiga.

OpenSearch Layanan secara otomatis menjadwalkan pembaruan ini dan memberi tahu Anda 72 jam (tiga hari) sebelum pembaruan yang dijadwalkan melalui email dan pemberitahuan konsol. Anda dapat memilih untuk segera menerapkan pembaruan atau menjadwalkannya kembali untuk tanggal dan waktu yang lebih tepat dalam jangka waktu yang diizinkan. Anda juga dapat menjadwalkannya selama jendela off-peak domain berikutnya. Jika Anda tidak mengambil tindakan pada pembaruan yang diperlukan dan Anda tidak membuat perubahan domain apa pun yang menyebabkan penyebaran biru/hijau, OpenSearch Layanan dapat memulai pembaruan kapan saja di luar batas waktu yang ditentukan (biasanya 14 hari sejak ketersediaan), di dalam jendela luar puncak domain.

Terlepas dari kapan pembaruan dijadwalkan, jika Anda membuat perubahan pada domain yang menyebabkan penyebaran biru/hijau, OpenSearch Layanan secara otomatis memperbarui domain Anda untuk Anda.

Pembaruan tambalan

Versi perangkat lunak layanan yang diakhiri dengan “-P” dan angka, seperti R20211203- *P4*, adalah rilis patch. Patch cenderung mencakup peningkatan kinerja, perbaikan bug kecil, dan perbaikan keamanan atau perbaikan postur. Rilis patch tidak menyertakan fitur baru atau perubahan yang melanggar, dan umumnya tidak memiliki dampak langsung atau nyata pada pengguna. Pemberitahuan perangkat lunak layanan memberi tahu Anda jika rilis patch bersifat opsional atau wajib.

Pertimbangan

Pertimbangkan hal berikut saat memutuskan apakah akan memperbarui domain Anda:

- Memperbarui domain Anda secara manual memungkinkan Anda memanfaatkan fitur baru dengan lebih cepat. Ketika Anda memilih Perbarui, OpenSearch Layanan menempatkan permintaan dalam antrian dan memulai pembaruan ketika ada waktu.
- Saat Anda memulai pembaruan perangkat lunak layanan, OpenSearch Layanan mengirimkan pemberitahuan saat pembaruan dimulai dan kapan selesai.
- Pembaruan perangkat lunak menggunakan penerapan biru/hijau untuk meminimalkan waktu henti. Pembaruan untuk sementara dapat menegangkan simpul utama terdedikasi klaster, jadi pastikan untuk mempertahankan kapasitas yang cukup untuk menangani overhead terkait.
- Pembaruan biasanya selesai dalam beberapa menit, tetapi juga dapat memakan waktu beberapa jam atau bahkan berhari-hari jika sistem Anda mengalami beban berat. Pertimbangkan untuk memperbarui domain Anda selama jendela off-peak yang dikonfigurasi untuk menghindari periode pembaruan yang lama.

Memulai pembaruan perangkat lunak layanan

Anda dapat meminta pembaruan perangkat lunak layanan melalui konsol OpenSearch Layanan AWS CLI, atau salah satu SDK.

Konsol

Untuk meminta pembaruan perangkat lunak layanan

1. Buka konsol OpenSearch Layanan Amazon di https://console.aws.amazon.com/aos/home.
2. Pilih nama domain untuk membuka konfigurasinya.
3. Pilih Tindakan, Perbarui dan pilih salah satu opsi berikut:

   - Terapkan pembaruan sekarang - Segera jadwalkan tindakan yang akan terjadi pada jam saat ini jika ada kapasitas yang tersedia. Jika kapasitas tidak tersedia, kami menyediakan slot waktu lain yang tersedia untuk dipilih.
   - Jadwalkan di jendela off-peak — Hanya tersedia jika jendela off-peak diaktifkan untuk domain. Menjadwalkan pembaruan yang akan dilakukan selama jendela off-peak domain yang dikonfigurasi. Tidak ada jaminan bahwa pembaruan akan terjadi selama jendela langsung berikutnya. Tergantung pada kapasitas, itu mungkin terjadi di hari-hari berikutnya. Untuk informasi selengkapnya, lihat the section called “Jendela off-peak”.
   - Jadwal untuk tanggal dan waktu tertentu - Menjadwalkan pembaruan untuk berlangsung pada tanggal dan waktu tertentu. Jika waktu yang Anda tentukan tidak tersedia karena alasan kapasitas, Anda dapat memilih slot waktu yang berbeda.

   Jika Anda menjadwalkan pembaruan di kemudian hari (di dalam atau di luar jendela off-peak domain), Anda dapat menjadwal ulang kapan saja. Untuk petunjuk, lihat the section called “Tindakan penjadwalan ulang”.
4. Pilih Konfirmasi.

AWS CLI

Kirim start-service-software-update AWS CLI permintaan untuk memulai pembaruan perangkat lunak layanan. Contoh ini segera menambahkan pembaruan ke antrian:

```
aws opensearch start-service-software-update \
  --domain-name my-domain \
  --schedule-at "NOW"
```

Tanggapan:

```
{
```

```
    "ServiceSoftwareOptions": {
        "CurrentVersion": "R20220928-P1",
        "NewVersion": "R20220928-P2",
        "UpdateAvailable": true,
        "Cancellable": true,
        "UpdateStatus": "PENDING_UPDATE",
        "Description": "",
        "AutomatedUpdateDate": "1969-12-31T16:00:00-08:00",
        "OptionalDeployment": true
    }
}
```

> **Tip**
>
> Setelah Anda meminta pembaruan, Anda memiliki jendela waktu yang sempit di mana Anda dapat membatalkannya. Durasi `PENDING_UPDATE` status ini dapat sangat bervariasi dan tergantung pada Anda Wilayah AWS dan jumlah pembaruan bersamaan yang dilakukan OpenSearch Layanan. Untuk membatalkan pembaruan, gunakan konsol atau `cancel-service-software-update` AWS CLI perintah.

Jika permintaan gagal dengan a`BaseException`, itu berarti bahwa waktu yang Anda tentukan tidak tersedia karena alasan kapasitas, dan Anda harus menentukan waktu yang berbeda. OpenSearch Layanan memberikan saran slot alternatif yang tersedia sebagai tanggapan.

AWS SDK

Contoh skrip Python ini menggunakan metode describe_domain dan start_service_software_update dari AWS SDK for Python (Boto3) untuk memeriksa apakah domain memenuhi syarat untuk pembaruan perangkat lunak layanan dan jika demikian, mulai pembaruan. Anda harus memberikan nilai untuk`domain_name`.

```
import boto3
from botocore.config import Config
import time

# Build the client using the default credential configuration.
# You can use the CLI and run 'aws configure' to set access key, secret
# key, and default region.

my_config = Config(
```

```
    # Optionally lets you specify a Region other than your default.
    region_name='us-east-1'
)

domain_name = ''  # The name of the domain to check and update

client = boto3.client('opensearch', config=my_config)


def getUpdateStatus(client):
    """Determines whether the domain is eligible for an update"""
    response = client.describe_domain(
        DomainName=domain_name
    )
    sso = response['DomainStatus']['ServiceSoftwareOptions']
    if sso['UpdateStatus'] == 'ELIGIBLE':
        print('Domain [' + domain_name + '] is eligible for a service software update
 from version ' +
              sso['CurrentVersion'] + ' to version ' + sso['NewVersion'])
        updateDomain(client)
    else:
        print('Domain is not eligible for an update at this time.')


def updateDomain(client):
    """Starts a service software update for the eligible domain"""
    response = client.start_service_software_update(
        DomainName=domain_name
    )
    print('Updating domain [' + domain_name + '] to version ' +
          response['ServiceSoftwareOptions']['NewVersion'] + '...')
    waitForUpdate(client)


def waitForUpdate(client):
    """Waits for the domain to finish updating"""
    response = client.describe_domain(
        DomainName=domain_name
    )
    status = response['DomainStatus']['ServiceSoftwareOptions']['UpdateStatus']
    if status == 'PENDING_UPDATE' or status == 'IN_PROGRESS':
        time.sleep(30)
        waitForUpdate(client)
    elif status == 'COMPLETED':
```

```
        print('Domain [' + domain_name +
              '] successfully updated to the latest software version')
    else:
        print('Domain is not currently being updated.')

def main():
    getUpdateStatus(client)
```

Menjadwalkan pembaruan perangkat lunak selama jendela off-peak

Setiap domain OpenSearch Layanan yang dibuat setelah 16 Februari 2023 memiliki jendela 10 jam harian antara pukul 22:00 dan 8:00 pagi waktu setempat yang kami anggap sebagai jendela off-peak. OpenSearch Layanan menggunakan jendela ini untuk menjadwalkan pembaruan perangkat lunak layanan untuk domain. Pembaruan off-peak membantu meminimalkan ketegangan pada node master khusus cluster selama periode lalu lintas yang lebih tinggi. OpenSearch Layanan tidak dapat memulai pembaruan di luar jendela 10 jam ini tanpa persetujuan Anda.

- Untuk pembaruan opsional, OpenSearch Layanan memberi tahu Anda tentang ketersediaan pembaruan dan meminta Anda untuk menjadwalkan pembaruan selama jendela off-peak yang akan datang.
- Untuk pembaruan yang diperlukan, OpenSearch Layanan secara otomatis menjadwalkan pembaruan selama jendela off-peak yang akan datang dan memberi tahu Anda tiga hari sebelumnya. Anda dapat menjadwal ulang pembaruan (untuk di dalam atau di luar jendela off-peak), tetapi hanya dalam jangka waktu yang diperlukan agar pembaruan selesai.

Untuk setiap domain, Anda dapat memilih untuk mengganti waktu mulai 10:00 P.M. default dengan waktu khusus. Untuk petunjuk, lihat the section called “Mengonfigurasi jendela off-peak khusus”.

Konsol

Untuk menjadwalkan pembaruan selama jendela off-peak yang akan datang

1. Buka konsol OpenSearch Layanan Amazon di https://console.aws.amazon.com/aos/home.
2. Pilih nama domain untuk membuka konfigurasinya.
3. Pilih Tindakan, Perbarui.
4. Pilih Jadwalkan di jendela off-peak.
5. Pilih Konfirmasi.

Anda dapat melihat tindakan terjadwal pada tab jendela Off-peak dan menjadwal ulang kapan saja. Lihat the section called “Melihat tindakan terjadwal”.

CLI

Untuk menjadwalkan pembaruan selama jendela off-peak mendatang menggunakan AWS CLI, kirim StartServiceSoftwareUpdatepermintaan dan tentukan `OFF_PEAK_WINDOW` `--schedule-at` parameternya:

```
aws opensearch start-service-software-update \
  --domain-name my-domain \
  --schedule-at "OFF_PEAK_WINDOW"
```

Pemantauan pembaruan perangkat lunak layanan

OpenSearch Layanan mengirimkan pemberitahuan ketika pembaruan perangkat lunak layanan tersedia, diperlukan, dimulai, selesai, atau gagal. Anda dapat melihat notifikasi ini di panel Notifikasi konsol OpenSearch Layanan. Kepelikan notifikasi adalah `Informational` jika pembaruan bersifat opsional dan `High` jika diperlukan.

OpenSearch Layanan juga mengirimkan acara perangkat lunak layanan ke Amazon EventBridge. Anda dapat menggunakan EventBridge untuk mengonfigurasi aturan yang mengirim email atau melakukan tindakan tertentu saat acara diterima. Untuk panduan contoh, lihat the section called “Tutorial: Mengirim peringatan SNS untuk pembaruan yang tersedia”.

Untuk melihat format setiap peristiwa perangkat lunak layanan yang dikirim ke Amazon EventBridge, lihatthe section called “Peristiwa pembaruan perangkat lunak layanan”.

Ketika domain tidak memenuhi syarat untuk pembaruan

Domain Anda tidak memenuhi syarat untuk pembaruan perangkat lunak layanan jika berada di salah satu status berikut:

Status	Deskripsi
Domain dalam pemrosesan	Domain sedang di tengah-tengah perubahan konfigurasi. Periksa kelayakan pembaruan setelah operasi selesai.
Status klaster merah	Satu atau lebih indeks di cluster berwarna merah. Untuk langkah-langkah pemecahan masalah, lihat the section called “Status klaster merah”.

Status	Deskripsi
Tingkat kesalahan tinggi	OpenSearch Cluster mengembalikan sejumlah besar kesalahan 5 xx saat mencoba memproses permintaan. Masalah ini biasanya merupakan hasil dari terlalu banyaknya permintaan baca atau tulis secara bersamaan. Pertimbangkan untuk mengurangi lalu lintas ke klaster atau menskalakan domain Anda.
Split brain	Otak terbelah berarti OpenSearch cluster Anda memiliki lebih dari satu simpul master dan telah dibagi menjadi dua cluster yang tidak akan pernah bergabung kembali dengan sendirinya. Anda dapat menghindari split brain dengan menggunakan jumlah yang direkomendasikan untuk simpul utama terdedikasi. Untuk bantuan pemulihan dari split brain, hubungi AWS Support.
Masalah integrasi Amazon Cognito	Domain Anda menggunakan autentikasi untuk OpenSearch Dasbor, dan OpenSearch Layanan tidak dapat menemukan satu atau beberapa sumber daya Amazon Cognito. Masalah ini biasanya terjadi jika kolam pengguna Amazon Cognito hilang. Untuk memperbaiki masalah, buat ulang sumber daya yang hilang dan konfigurasikan domain OpenSearch Layanan untuk menggunakannya.
Masalah layanan lainnya	Masalah dengan OpenSearch Layanan itu sendiri dapat menyebabk an domain Anda ditampilkan sebagai tidak memenuhi syarat untuk pembaruan. Jika tidak ada syarat sebelumnya yang berlaku untuk domain Anda dan masalah berlanjut selama lebih dari satu hari, hubungi AWS Support.

Mendefinisikan jendela off-peak untuk Amazon Service OpenSearch

Saat membuat domain OpenSearch Layanan Amazon, Anda menentukan jendela 10 jam harian yang dianggap jam sibuk. OpenSearch Layanan menggunakan jendela ini untuk menjadwalkan pembaruan perangkat lunak layanan dan pengoptimalan Auto-Tune yang memerlukan penerapan biru/hijau selama waktu lalu lintas yang relatif lebih rendah, bila memungkinkan. Biru/hijau mengacu

pada proses menciptakan lingkungan baru untuk pembaruan domain dan mengarahkan pengguna ke lingkungan baru setelah pembaruan tersebut selesai.

Meskipun penerapan biru/hijau tidak mengganggu, untuk meminimalkan potensi dampak kinerja saat sumber daya dikonsumsi untuk penerapan biru/hijau, sebaiknya Anda menjadwalkan penerapan ini selama jendela off-peak domain yang dikonfigurasi. Pembaruan seperti penggantian node, atau yang perlu segera disebarkan ke domain, jangan gunakan jendela off-peak.

Anda dapat memodifikasi waktu mulai untuk jendela off-peak, tetapi Anda tidak dapat mengubah panjang jendela.

> **Note**
>
> Jendela off-peak diperkenalkan pada 16 Februari 2023. Semua domain yang dibuat sebelum tanggal ini memiliki jendela off-peak dinonaktifkan secara default. Anda harus mengaktifkan dan mengonfigurasi jendela off-peak secara manual untuk domain ini. Semua domain yang dibuat setelah tanggal ini akan mengaktifkan jendela off-peak secara default. Anda tidak dapat menonaktifkan jendela off-peak untuk domain setelah diaktifkan.

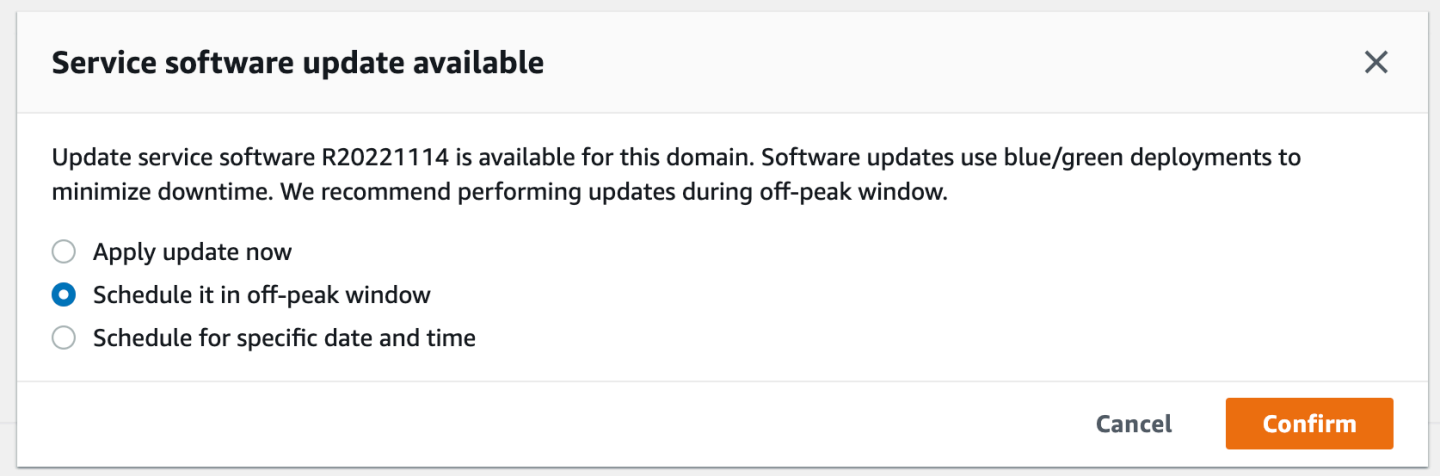
Topik

Pembaruan perangkat lunak layanan off-peak

OpenSearch Layanan memiliki dua kategori pembaruan perangkat lunak layanan yang luas - opsional dan wajib. Kedua jenis membutuhkan penerapan biru/hijau. Pembaruan opsional tidak diberlakukan pada domain Anda, sementara pembaruan yang diperlukan diinstal secara otomatis jika Anda tidak mengambil tindakan sebelum batas waktu yang ditentukan (biasanya dua minggu sejak

ketersediaan). Untuk informasi selengkapnya, lihat the section called “Pembaruan opsional versus yang diperlukan”.

Saat Anda memulai pembaruan opsional, Anda memiliki pilihan untuk segera menerapkan pembaruan, menjadwalkannya untuk jendela off-peak berikutnya, atau menentukan tanggal dan waktu khusus untuk menerapkannya.



Untuk pembaruan yang diperlukan, OpenSearch Layanan secara otomatis menjadwalkan tanggal dan waktu selama jam sibuk untuk melakukan pembaruan. Anda menerima pemberitahuan tiga hari sebelum pembaruan yang dijadwalkan, dan Anda dapat memilih untuk menjadwal ulang untuk tanggal dan waktu berikutnya dalam periode penerapan yang diperlukan. Untuk petunjuk, lihat the section called “Tindakan penjadwalan ulang”.

Optimasi Auto-Tune Off-peak

Sebelumnya, Auto-Tune menggunakan jendela pemeliharaan untuk menjadwalkan perubahan yang memerlukan penerapan biru/hijau. Domain yang sudah mengaktifkan Auto-Tune dan jendela pemeliharaan sebelum pengenalan jendela off-peak akan terus menggunakan jendela pemeliharaan untuk pembaruan ini, kecuali jika Anda memigrasikannya untuk menggunakan jendela off-peak.

Kami menyarankan Anda memigrasikan domain Anda untuk menggunakan jendela off-peak, karena digunakan untuk menjadwalkan aktivitas lain di domain seperti pembaruan softwate layanan. Untuk petunjuk, lihat the section called “Migrasi dari jendela pemeliharaan Auto-Tune”. Anda tidak dapat kembali menggunakan jendela pemeliharaan setelah memigrasikan domain ke jendela off-peak.

Semua domain yang dibuat setelah 16 Februari 2023 akan menggunakan jendela off-peak, bukan jendela pemeliharaan lama, untuk menjadwalkan penerapan biru/hijau. Anda tidak dapat

menonaktifkan jendela off-peak untuk domain. Untuk daftar pengoptimalan Auto-Tune yang memerlukan penerapan biru/hijau, lihat. the section called “Jenis perubahan”

Mengaktifkan jendela off-peak

Setiap domain yang dibuat sebelum 16 Februari 2023 (ketika jendela off-peak diperkenalkan) menonaktifkan fitur tersebut secara default. Anda harus mengaktifkannya secara manual untuk domain ini. Anda tidak dapat menonaktifkan jendela off-peak setelah diaktifkan.

Konsol

Untuk mengaktifkan jendela off-peak untuk domain

1. Buka konsol OpenSearch Layanan Amazon di https://console.aws.amazon.com/aos/home.
2. Pilih nama domain untuk membuka konfigurasinya.
3. Arahkan ke tab jendela Off-peak dan pilih Edit.
4. Tentukan waktu mulai khusus di Coordinated Universal Time (UTC). Misalnya, untuk mengonfigurasi waktu mulai pukul 11:30 P.M. di Wilayah AS Barat (Oregon), tentukan 07:30.
5. Pilih Smpan Perubahan.

CLI

Untuk memodifikasi jendela off-peak menggunakanAWS CLI, kirim UpdateDomainConfigpermintaan:

```
aws opensearch update-domain-config \
  --domain-name my-domain \
  --off-peak-window-options 'Enabled=true,
 OffPeakWindow={WindowStartTime={Hours=02,Minutes=00}}'
```

Jika Anda tidak menentukan waktu mulai jendela kustom, defaultnya adalah 00:00 UTC.

Mengonfigurasi jendela off-peak khusus

Anda menentukan jendela off-peak khusus untuk domain Anda di Coordinated Universal Time (UTC). Misalnya, jika Anda ingin jendela off-peak dimulai pada pukul 11:00 P.M. untuk domain di Wilayah AS Timur (Virginia Utara), Anda akan menentukan pukul 04:00 UTC.

Konsol

Untuk memodifikasi jendela off-peak untuk domain

1. Buka konsol OpenSearch Layanan Amazon di https://console.aws.amazon.com/aos/home.
2. Pilih nama domain untuk membuka konfigurasinya.
3. Arahkan ke tab jendela Off-peak. Anda dapat melihat jendela off-peak yang dikonfigurasi dan daftar tindakan terjadwal yang akan datang untuk domain.
4. Pilih Edit dan tentukan waktu mulai baru di UTC. Misalnya, untuk mengkonfigurasi waktu mulai 9:00 PM di Wilayah AS Timur (Virginia N.), tentukan 02:00 UCT.
5. Pilih Smpan Perubahan.

CLI

Untuk mengonfigurasi jendela off-peak khusus menggunakanAWS CLI, kirim UpdateDomainConfigpermintaan dan tentukan jam dan menit dalam format waktu 24 jam.

Misalnya, permintaan berikut mengubah waktu mulai jendela menjadi 2:00 AM UTC:

```
aws opensearch update-domain-config \
  --domain-name my-domain \
  --off-peak-window-options 'OffPeakWindow={WindowStartTime={Hours=02,Minutes=00}}'
```

Jika Anda tidak menentukan waktu mulai jendela, defaultnya adalah 10:00 P.M. waktu setempat untuk domain Wilayah AWS tersebut dibuat.

Melihat tindakan terjadwal

Anda dapat melihat semua tindakan yang saat ini dijadwalkan, sedang berlangsung, atau tertunda untuk setiap domain Anda. Tindakan dapat memiliki tingkat keparahan`HIGH`,`MEDIUM`, dan`LOW`.

Tindakan dapat memiliki status berikut:

- `Pending update`— Tindakan dalam antrian untuk diproses.
- `In progress`— Tindakan saat ini sedang berlangsung.
- `Failed`— Tindakan gagal diselesaikan.
- `Completed`— Tindakan telah selesai dengan sukses.

- `Not eligible`— Hanya untuk pembaruan perangkat lunak layanan. Pembaruan tidak dapat dilanjutkan karena cluster dalam keadaan tidak sehat.
- `Eligible`— Hanya untuk pembaruan perangkat lunak layanan. Domain memenuhi syarat untuk pembaruan.

Konsol

Konsol OpenSearch Layanan menampilkan semua tindakan terjadwal dalam konfigurasi domain, bersama dengan tingkat keparahan setiap tindakan dan status saat ini.

Untuk melihat tindakan terjadwal untuk domain

1. Buka konsol OpenSearch Layanan Amazon di https://console.aws.amazon.com/aos/home.
2. Pilih nama domain untuk membuka konfigurasinya.
3. Arahkan ke tab jendela Off-peak.
4. Di bawah Tindakan terjadwal, lihat semua tindakan yang saat ini dijadwalkan, sedang berlangsung, atau tertunda untuk domain.

CLI

Untuk melihat tindakan terjadwal menggunakanAWS CLI, kirim ListScheduledActionspermintaan:

```
aws opensearch list-scheduled-actions \
  --domain-name my-domain
```

Tanggapan:

```
{
    "ScheduledActions": [
        {
            "Cancellable": true,
            "Description": "The Deployment type is : BLUE_GREEN.",
            "ID": "R20220721-P13",
            "Mandatory": false,
            "Severity": "HIGH",
            "ScheduledBy": "CUSTOMER",
            "ScheduledTime": 1.673871601E9,
            "Status": "PENDING_UPDATE",
```

```
            "Type": "SERVICE_SOFTWARE_UPDATE",
        },
        {
            "Cancellable": true,
            "Description": "Amazon Opensearch will adjust the young generation JVM
 arguments on your domain to improve performance",
            "ID": "Auto-Tune",
            "Mandatory": true,
            "Severity": "MEDIUM",
            "ScheduledBy": "SYSTEM",
            "ScheduledTime": 1.673871601E9,
            "Status": "PENDING_UPDATE",
            "Type": "JVM_HEAP_SIZE_TUNING",
        }
    ]
}
```

Tindakan penjadwalan ulang

OpenSearch Layanan memberi tahu Anda tentang pembaruan perangkat lunak layanan terjadwal dan pengoptimalan Auto-Tune. Anda dapat memilih untuk segera menerapkan perubahan, atau menjadwalkannya kembali untuk tanggal dan waktu nanti.

> **Note**
>
> OpenSearch Layanan dapat menjadwalkan tindakan dalam waktu satu jam dari waktu yang Anda pilih. Misalnya, jika Anda memilih untuk menerapkan pembaruan pada jam 5 sore, itu dapat diterapkan antara jam 5 dan 6 sore.

Konsol

Untuk menjadwal ulang suatu tindakan

1. Buka konsol OpenSearch Layanan Amazon di https://console.aws.amazon.com/aos/home.
2. Pilih nama domain untuk membuka konfigurasinya.
3. Arahkan ke tab jendela Off-peak.
4. Di bawah Tindakan terjadwal, pilih tindakan dan pilih Reschedule.
5. Pilih salah satu opsi berikut:

- Terapkan pembaruan sekarang - Segera jadwalkan tindakan yang akan terjadi pada jam saat ini jika ada kapasitas yang tersedia. Jika kapasitas tidak tersedia, kami menyediakan slot waktu lain yang tersedia untuk dipilih.
- Jadwalkan di jendela off-peak - Menandai aksi yang akan diambil selama jendela off-peak yang akan datang. Tidak ada jaminan bahwa perubahan akan diterapkan selama jendela berikutnya. Tergantung pada kapasitas, itu mungkin terjadi di hari-hari berikutnya.
- Jadwalkan ulang pembaruan ini - Memungkinkan Anda menentukan tanggal dan waktu khusus untuk menerapkan perubahan. Jika waktu yang Anda tentukan tidak tersedia karena alasan kapasitas, Anda dapat memilih slot waktu yang berbeda.
- Batalkan pembaruan terjadwal - Membatalkan pembaruan. Opsi ini hanya tersedia untuk pembaruan perangkat lunak layanan opsional. Ini tidak tersedia untuk tindakan Auto-Tune atau pembaruan perangkat lunak wajib.

6. Pilih Smpan Perubahan.

CLI

Untuk menjadwal ulang tindakan menggunakanAWS CLI, kirim UpdateScheduledActionpermintaan. Untuk mengambil ID tindakan, kirim `ListScheduledActions` permintaan.

Permintaan berikut menjadwalkan ulang pembaruan perangkat lunak layanan untuk tanggal dan waktu tertentu:

```
aws opensearch update-scheduled-action \
  --domain-name my-domain \
  --action-id R20220721-P13 \
  --action-type "SERVICE_SOFTWARE_UPDATE" \
  --desired-start-time 1677348395000 \
  --schedule-at TIMESTAMP
```

Tanggapan:

```
{
   "ScheduledAction": {
      "Cancellable": true,
      "Description": "Cluster status is updated.",
      "Id": "R20220721-P13",
      "Mandatory": false,
      "ScheduledBy": "CUSTOMER",
```

```
      "ScheduledTime": 1677348395000,
      "Severity": "HIGH",
      "Status": "PENDING_UPDATE",
      "Type": "SERVICE_SOFTWARE_UPDATE"
   }
}
```

Jika permintaan gagal dengan a`SlotNotAvailableException`, itu berarti bahwa waktu yang Anda tentukan tidak tersedia karena alasan kapasitas, dan Anda harus menentukan waktu yang berbeda. OpenSearch Layanan memberikan saran slot alternatif yang tersedia sebagai tanggapan.

Migrasi dari jendela pemeliharaan Auto-Tune

Jika domain dibuat sebelum 16 Februari 2023, domain dapat menggunakan jendela pemeliharaan untuk menjadwalkan pengoptimalan Auto-Tune yang memerlukan penerapan biru/hijau. Anda dapat memigrasikan domain Auto-Tune yang ada untuk menggunakan jendela off-peak sebagai gantinya.

> **Note**
>
> Anda tidak dapat kembali menggunakan jendela pemeliharaan setelah memigrasikan domain untuk menggunakan jendela off-peak.

Konsol

Untuk memigrasikan domain untuk menggunakan jendela off-peak

1. Di dalam konsol OpenSearch Layanan Amazon, pilih nama domain untuk membuka konfigurasinya.
2. Buka tab Auto-Tune dan pilih Edit.
3. Pilih Migrasikan ke jendela off-peak.
4. Untuk Waktu mulai (UTC), sediakan waktu mulai harian untuk jendela off-peak di Universal Coordinated Time (UTC).
5. Pilih Smpan Perubahan.

CLI

Untuk bermigrasi dari jendela pemeliharaan Auto-Tune ke jendela off-peak menggunakanAWS CLI, kirim permintaan: UpdateDomainConfig

```
aws opensearch update-domain-config \
  --domain-name my-domain \
  --auto-tune-options
 DesiredState=ENABLED,UseOffPeakWindow=true,MaintenanceSchedules=[]
```

Jendela off-peak harus dihidupkan agar Anda dapat memigrasikan domain dari jendela pemeliharaan Auto-Tune ke jendela off-peak. Anda dapat mengaktifkan jendela off-peak dalam permintaan terpisah atau dalam permintaan yang sama. Untuk petunjuk, lihat the section called “Mengaktifkan jendela off-peak”.

Pemberitahuan di OpenSearch Layanan Amazon

Pemberitahuan di OpenSearch Layanan Amazon berisi informasi penting tentang kinerja dan kesehatan domain Anda. OpenSearch Layanan memberi tahu Anda tentang pembaruan perangkat lunak layanan, penyempurnaan Auto-Tune, peristiwa kesehatan klaster, dan kesalahan domain. Pemberitahuan tersedia untuk semua versi OpenSearch dan Elasticsearch OSS.

Anda dapat melihat notifikasi di panel Pemberitahuan konsol OpenSearch Layanan. Semua pemberitahuan untuk OpenSearch Layanan juga muncul di Amazon EventBridge. Untuk daftar lengkap notifikasi dan contoh peristiwa, lihatthe section called “Pemantauan peristiwa”.

Topik

- Memulai dengan notifikasi
- Notifikasi kepelikan
- Contoh EventBridge acara

Memulai dengan notifikasi

Notifikasi diaktifkan secara otomatis saat Anda membuat domain. Buka panel Pemberitahuan konsol OpenSearch Layanan untuk memantau dan mengakui pemberitahuan. Setiap notifikasi mencakup informasi seperti waktu notifikasi diposting, domain yang berhubungan, tingkat kepelikan dan status, dan penjelasan singkat. Anda dapat melihat notifikasi historis hingga 90 hari di konsol.

Setelah mengakses panel Notifikasi atau mengakui pemberitahuan, Anda mungkin menerima pesan galat tentang tidak memiliki izin untuk melakukan atau. `es:ListNotifications` `es:UpdateNotificationStatus` Untuk mengatasi masalah ini, berikan izin berikut kepada pengguna atau peran Anda di IAM:

```
{
  "Version": "2012-10-17",
  "Statement": [{
    "Effect": "Allow",
    "Action": [
      "es:UpdateNotificationStatus",
      "es:ListNotifications"
    ],
    "Resource": "arn:aws:es:*:123456789012:domain/*"
  }]
}
```

Konsol IAM memunculkan kesalahan (“IAM tidak mengenali satu atau lebih tindakan.”) yang dapat Anda abaikan dengan aman. Anda juga dapat membatasi `es:UpdateNotificationStatus` tindakan ke domain tertentu. Untuk mempelajari selengkapnya, lihat the section called “Referensi elemen kebijakan”.

Notifikasi kepelikan

Pemberitahuan dalam OpenSearch Layanan dapat bersifat informasi, yang terkait dengan tindakan apa pun yang telah Anda lakukan atau pengoperasian domain Anda, atau dapat ditindaklanjuti, yang mengharuskan Anda untuk mengambil tindakan spesifik seperti menerapkan patch keamanan wajib. Setiap pemberitahuan memiliki tingkat keparahan yang terkait dengannya, yang dapat berupa`Informational`,`Low`,`Medium`,`High`, atau`Critical`. Tabel berikut merangkum setiap tingkat keparahan:

Kepelikan	Deskripsi	Contoh
`Informational`	Informasi yang terkait dengan pengoperasian domain Anda.	• Tersedia pembaruan perangkat lunak layanan • Auto-Tune dimulai
`Low`	Tindakan yang disarankan, namun tidak berdampak buruk pada ketersediaan domain atau performa jika tidak ada tindakan yang diambil.	• Auto-Tune dibatalkan • Peringatan jumlah pecahan tinggi

Kepelikan	Deskripsi	Contoh
`Medium`	Mungkin ada dampak jika tindakan yang disaranka n tidak dilakukan, tetapi dilengkapi dengan jendela waktu yang diperpanjang untuk tindakan yang akan diambil.	• Pembaruan perangkat lunak layanan gagal • Batas jumlah pecahan terlampaui
`High`	Tindakan mendesak diperlukan untuk menghindari dampak buruk.	• Pembaruan perangkat lunak layanan diperlukan • Kunci KMS tidak dapat diakses
`Critical`	Tindakan segera diperluka n untuk menghindari dampak buruk, atau untuk pulih darinya.	Tidak ada yang tersedia

Contoh EventBridge acara

Contoh berikut menunjukkan acara pemberitahuan OpenSearch Layanan yang dikirim ke Amazon EventBridge. Pemberitahuan memiliki tingkat keparahan `Informational` karena pembaruan bersifat opsional:

```
{
  "version": "0",
  "id": "01234567-0123-0123-0123-012345678901",
  "detail-type": "Amazon OpenSearch Service Software Update Notification",
  "source": "aws.es",
  "account": "123456789012",
  "time": "2016-11-01T13:12:22Z",
  "region": "us-east-1",
  "resources": ["arn:aws:es:us-east-1:123456789012:domain/test-domain"],
  "detail": {
    "event": "Service Software Update",
    "status": "Available",
    "severity": "Informational",
```

```
    "description": "Service software update [R20200330-p1] available."
  }
}
```

Mengonfigurasi domain Multi-AZ di Amazon Service OpenSearch

Untuk mencegah kehilangan data dan meminimalkan downtime kluster Amazon OpenSearch Service jika terjadi gangguan layanan, Anda dapat mendistribusikan node di dua atau tiga Availability Zone di Region yang sama, konfigurasi yang dikenal sebagai Multi-AZ. Availability Zone adalah lokasi terisolasi di setiap AWS Wilayah.

Untuk domain yang menjalankan beban kerja produksi, kami merekomendasikan opsi penyebaran Multi-AZ dengan Siaga, yang membuat konfigurasi berikut:

- Domain dikerahkan di tiga zona.
- Jenis instance generasi saat ini untuk node master dan node data khusus.
- Tiga node master khusus dan tiga (atau kelipatan dari tiga) node data.
- Setidaknya dua replika untuk setiap indeks di domain Anda, atau kelipatan dari tiga salinan data (termasuk node primer dan replika).

Sisa bagian ini memberikan penjelasan dan konteks seputar konfigurasi ini.

Multi-AZ dengan Siaga

Multi-AZ dengan Standby adalah opsi penerapan untuk domain OpenSearch Layanan Amazon yang menawarkan ketersediaan 99,99%, kinerja yang konsisten untuk beban kerja produksi, serta konfigurasi dan manajemen domain yang disederhanakan. Saat Anda menggunakan Multi-AZ dengan Siaga, domain akan tahan terhadap kegagalan infrastruktur, tanpa berdampak pada kinerja atau ketersediaan. Opsi penerapan ini mencapai standar ini dengan mengamanatkan sejumlah praktik terbaik, seperti jumlah node data tertentu, jumlah node master, jenis instance, jumlah replika, pengaturan pembaruan perangkat lunak, dan Penyetelan Otomatis yang diaktifkan.

Saat Anda menggunakan Multi-AZ dengan Siaga, OpenSearch Layanan membuat domain di tiga Availability Zone, dengan setiap zona berisi salinan data lengkap dan dengan data yang didistribusikan secara merata di setiap zona. Domain Anda menyimpan node di salah satu zona ini sebagai siaga, yang berarti bahwa mereka tidak melayani permintaan pencarian. Ketika OpenSearch

Service mendeteksi kegagalan dalam infrastruktur yang mendasarinya, secara otomatis mengaktifkan node siaga dalam waktu kurang dari satu menit. Domain terus melayani permintaan pengindeksan dan pencarian, dan dampak apa pun terbatas pada waktu yang diperlukan untuk melakukan failover. Tidak ada redistribusi data atau sumber daya, yang menghasilkan kinerja cluster yang tidak terpengaruh dan tidak ada risiko ketersediaan yang menurun. Multi-AZ dengan Siaga tersedia tanpa biaya tambahan.

Anda memiliki dua opsi untuk membuat domain dengan standby di file. AWS Management Console Pertama, Anda dapat membuat domain dengan metode Easy create creation, dan OpenSearch Service akan secara otomatis menggunakan konfigurasi yang telah ditentukan, yang meliputi berikut ini:

- Tiga Availability Zone, dengan satu bertindak sebagai siaga
- Tiga node master dan node data khusus
- Auto-Tune diaktifkan pada domain
- Penyimpanan GP3 untuk node data

Anda juga dapat memilih metode pembuatan pembuatan Standar dan memilih Domain dengan standby sebagai opsi penerapan Anda. Ini memungkinkan Anda untuk menyesuaikan domain Anda sambil tetap mengamanatkan fitur utama siaga, seperti tiga zona dan tiga node master. Sebaiknya pilih jumlah node data yang kelipatan dari tiga (jumlah Availability Zones).

Setelah Anda membuat domain, Anda dapat menavigasi ke halaman detail domain dan, di tab konfigurasi Cluster, konfirmasikan bahwa 3-AZ dengan siaga muncul di bawah Availability Zone (s).

Jika Anda mengalami masalah dalam memigrasi domain yang ada ke Multi-AZ dengan Siaga, lihat Kesalahan bermigrasi ke Multi-AZ dengan Siaga di panduan pemecahan masalah.

Batasan

Saat Anda menyiapkan domain dengan Multi-AZ dengan Siaga, pertimbangkan batasan berikut:

- Jumlah total pecahan pada node tidak dapat melebihi 1000, jumlah total pecahan pada cluster tidak dapat melebihi 75000, dan ukuran pecahan tunggal tidak dapat melebihi 65 GB.
- Multi-AZ dengan Standby hanya berfungsi dengan tipe`m5`,`c5`,`r5`,`r6g`,`c6g`,`m6g`, `r6gd` dan `i3` instance. Untuk informasi selengkapnya tentang instance yang didukung, lihat Jenis instans yang didukung.

- Anda hanya dapat menggunakan Provisioned IOPS SSD, General Purpose SSD (GP3), atau penyimpanan yang didukung instans dengan standby.

Multi-AZ tanpa Siaga

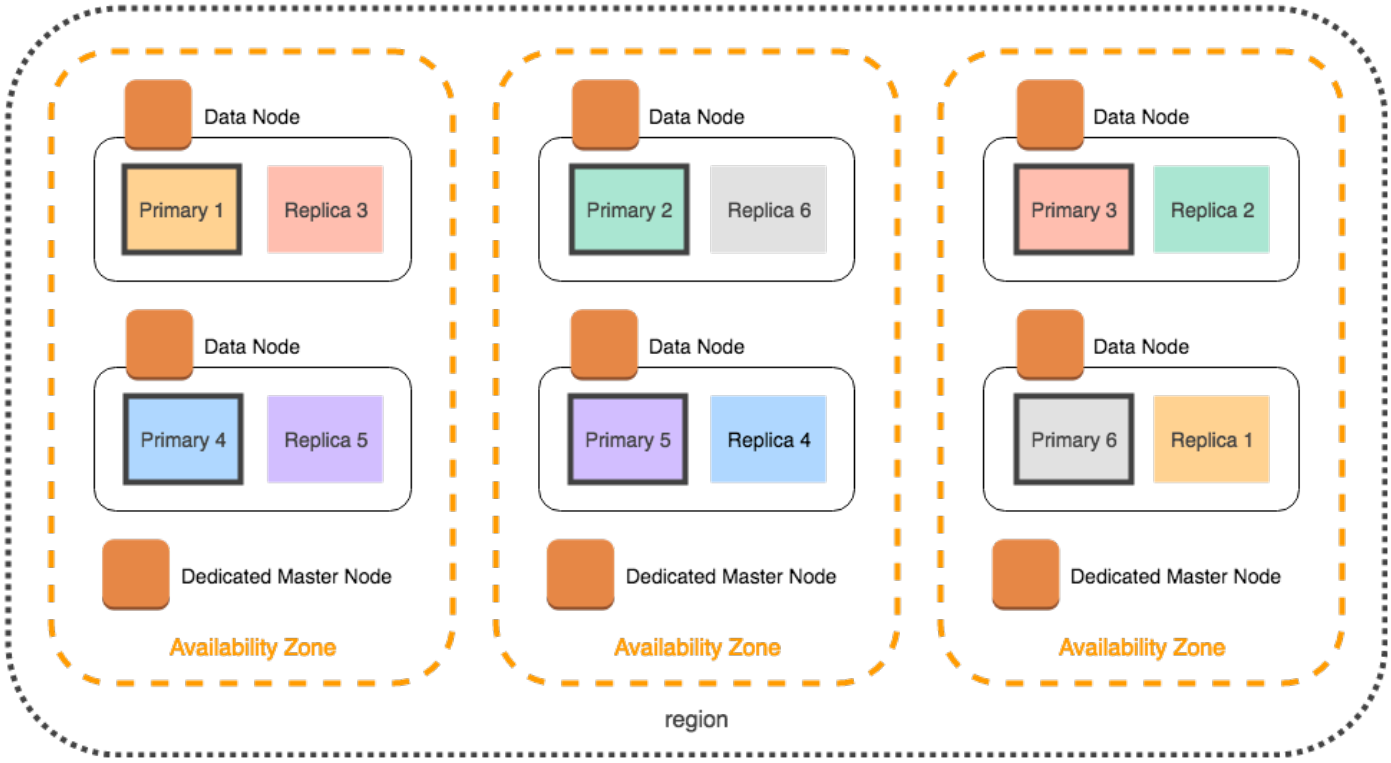
OpenSearch Layanan masih mendukung Multi-AZ tanpa Standby, yang menawarkan ketersediaan 99,9%. Node didistribusikan di seluruh Availability Zone (s), dan ketersediaan tergantung pada jumlah Availability Zone dan salinan data. Sedangkan dengan standby Anda harus mengonfigurasi domain Anda dengan praktik terbaik, tanpa siaga Anda dapat memilih jumlah Availability Zone, node, dan replika Anda sendiri. Kami tidak merekomendasikan opsi ini kecuali Anda memiliki alur kerja yang ada yang akan terganggu dengan membuat domain dengan siaga.

Jika Anda memilih opsi ini, kami tetap menyarankan Anda memilih tiga Availability Zone agar tetap tahan terhadap kegagalan node, disk, dan single-AZ. Ketika kegagalan terjadi, cluster mendistribusikan kembali data di seluruh sumber daya yang tersisa untuk menjaga ketersediaan dan redundansi. Pergerakan data ini meningkatkan penggunaan sumber daya pada cluster, dan dapat berdampak pada kinerja. Jika cluster tidak berukuran dengan benar, ia dapat mengalami penurunan ketersediaan, yang sebagian besar mengalahkan tujuan multi-AZ.

Satu-satunya cara untuk mengonfigurasi domain tanpa siaga AWS Management Console adalah dengan memilih metode pembuatan pembuatan Standar, dan pilih Domain tanpa siaga sebagai opsi penerapan Anda.

Distribusi serpihan

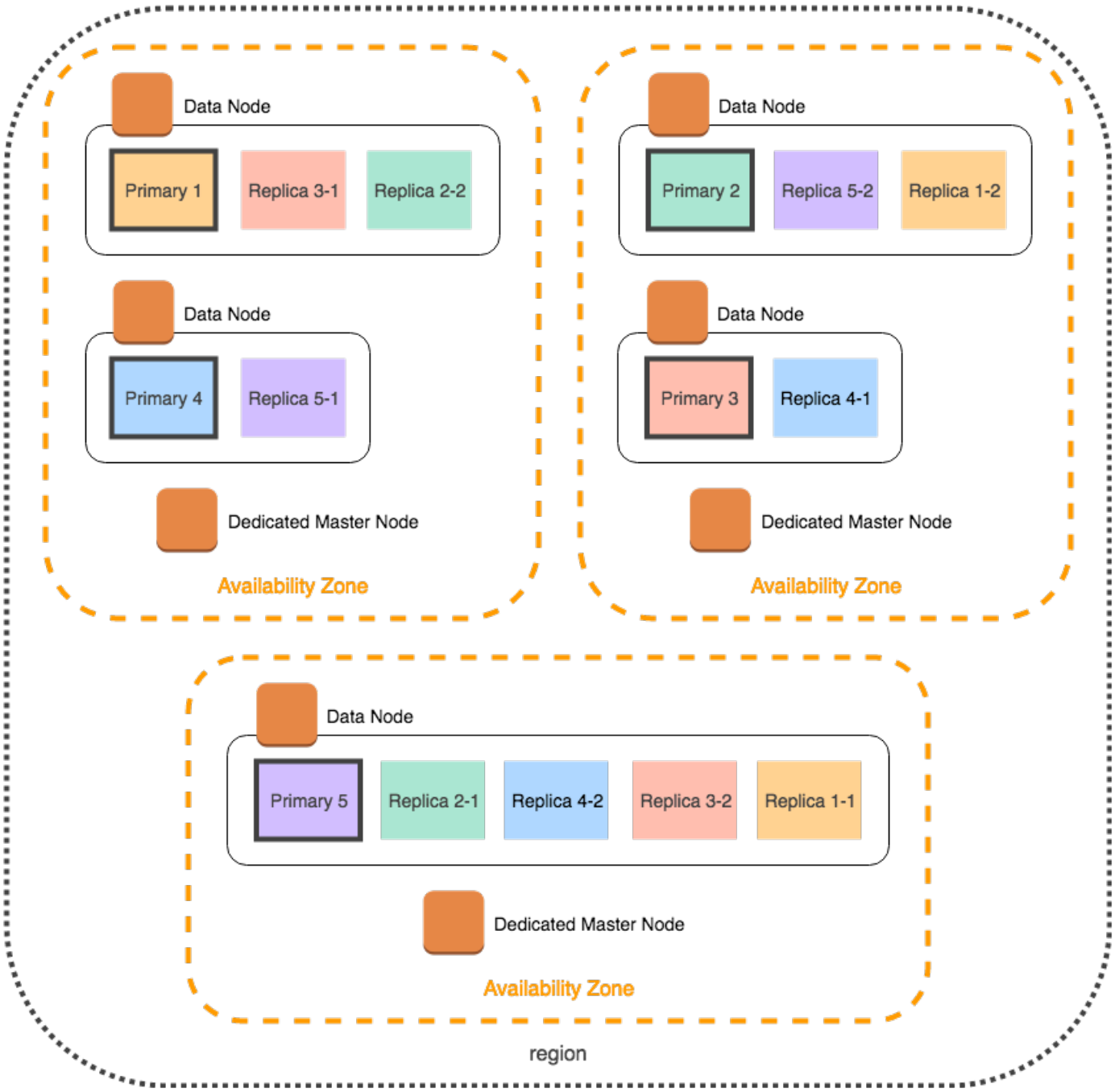
Jika Anda mengaktifkan Multi-AZ tanpa Standby, Anda harus membuat setidaknya satu replika untuk setiap indeks di cluster Anda. Tanpa replika, OpenSearch Layanan tidak dapat mendistribusikan salinan data Anda ke Availability Zone lainnya. Untungnya, konfigurasi default untuk setiap indeks adalah jumlah replika 1. Seperti yang ditunjukkan diagram berikut, OpenSearch Layanan melakukan upaya terbaik untuk mendistribusikan pecahan primer dan pecahan replika yang sesuai ke zona yang berbeda.



Selain mendistribusikan pecahan berdasarkan Availability Zone, OpenSearch Service mendistribusikannya berdasarkan node. Namun, konfigurasi domain tertentu dapat mengakibatkan jumlah serpihan tidak seimbang. Pertimbangkan domain berikut:

- 5 simpul data
- 5 serpihan primer
- 2 replika
- 3 Availability Zone

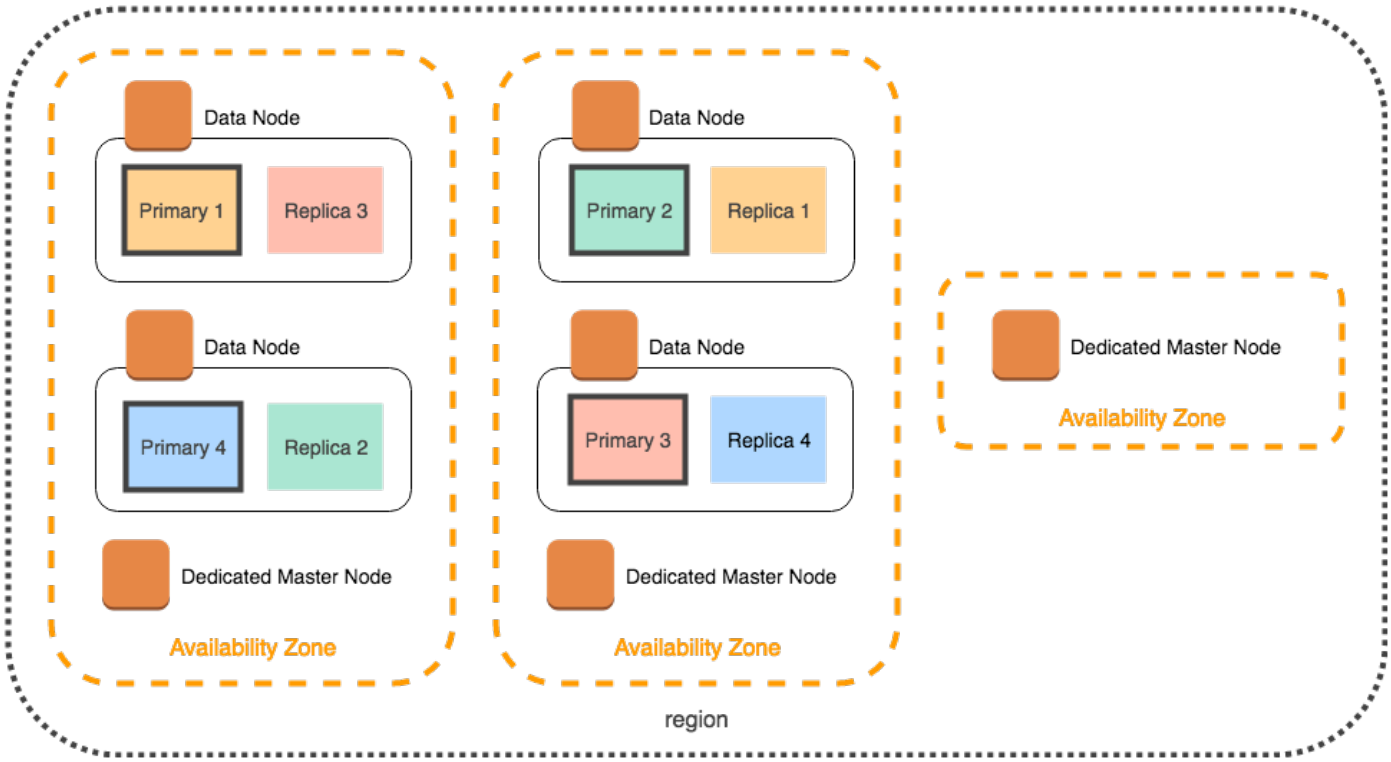
Dalam situasi ini, OpenSearch Service harus membebani satu node untuk mendistribusikan pecahan primer dan replika di seluruh zona, seperti yang ditunjukkan pada diagram berikut.



Untuk menghindari situasi seperti ini, yang dapat membebani node individual dan merusak kinerja, sebaiknya pilih Multi-AZ dengan Standby, atau pilih hitungan instans yang merupakan kelipatan tiga saat Anda berencana untuk memiliki dua atau lebih replika per indeks.

Distribusi simpul utama khusus

Bahkan jika Anda memilih dua Availability Zone saat mengonfigurasi domain Anda, OpenSearch Service secara otomatis mendistribusikan node master khusus di tiga Availability Zone. Distribusi ini membantu mencegah downtime klaster jika zona mengalami gangguan layanan. Jika Anda menggunakan tiga simpul utama khusus yang direkomendasikan dan satu Availability Zone tidak dapat digunakan, klaster Anda masih memiliki kuorum (2) simpul utama khusus dan dapat memilih master baru. Diagram berikut menunjukkan konfigurasi ini.



Jika Anda memilih tipe instans generasi lama yang tidak tersedia di tiga Availability Zone, skenario berikut berlaku:

- Jika Anda memilih tiga Availability Zone untuk domain, OpenSearch Service akan menampilkan error. Pilih tipe instans yang berbeda, dan coba lagi.
- Jika Anda memilih dua Availability Zone untuk domain, OpenSearch Service mendistribusikan node master khusus di dua zona.

Gangguan Availability Zone

Gangguan Availability Zone adalah hal yang jarang, namun bisa terjadi. Tabel berikut mencantumkan konfigurasi Multi-AZ yang berbeda dan perilaku selama gangguan. Baris terakhir dalam tabel berlaku untuk Multi-AZ dengan Standby, sementara semua baris lainnya memiliki konfigurasi yang hanya berlaku untuk Multi-AZ tanpa Standby.

Jumlah Availability Zone di suatu wilayah	Jumlah Availability Zone yang Anda pilih	Jumlah simpul utama khusus	Perilaku jika satu Availability Zone mengalami gangguan
2 atau lebih	2	0	Downtime. Klaster Anda kehilangan setengah dari simpul data dan harus mengganti setidaknya satu di Availability Zone yang tersisa sebelum dapat memilih master.
2	2	3	50/50 kemungkinan downtime. OpenSearch Layanan mendistribusikan dua node master khusus ke dalam satu Availability Zone dan satu ke yang lain: • Jika Availability Zone dengan satu simpul utama khusus mengalami gangguan, dua simpul utama khusus di Availability Zone yang tersisa dapat memilih master. • Jika Availability Zone dengan dua simpul utama khusus mengalami gangguan, klaster ini tidak tersedia sampai Availability Zone yang tersisa pulih.
3 atau lebih	2	3	Tidak ada downtime. OpenSearch Layanan secara otomatis mendistribusikan node master khusus di tiga Availability Zone, sehingga dua node master khusus yang tersisa dapat memilih master.

Jumlah Availability Zone di suatu wilayah	Jumlah Availability Zone yang Anda pilih	Jumlah simpul utama khusus	Perilaku jika satu Availability Zone mengalami gangguan
3 atau lebih	3	0	Tidak ada downtime. Kira-kira dua pertiga dari simpul data Anda masih tersedia untuk memilih master.
3 atau lebih	3	3	Tidak ada downtime. Sisanya dua simpul utama khusus dapat memilih master.

Dalam semua konfigurasi, terlepas dari penyebabnya, kegagalan node dapat menyebabkan node data cluster yang tersisa mengalami periode peningkatan beban sementara OpenSearch Service secara otomatis mengonfigurasi node baru untuk menggantikan node yang sekarang hilang.

Sebagai contoh, jika terjadi gangguan Availability Zone dalam konfigurasi tiga zona, dua-pertiga simpul data harus memproses hanya sebagai banyak permintaan untuk klaster. Ketika mereka memproses permintaan ini, simpul yang tersisa juga mereplikasi pecahan ke simpul baru karena mereka datang online, yang dapat lebih mempengaruhi performa. Jika ketersediaan sangat penting untuk beban kerja Anda, pertimbangkan untuk menambahkan sumber daya ke klaster Anda untuk mengurangi kekhawatiran ini.

> **Note**
>
> OpenSearch Layanan mengelola domain Multi-AZ secara transparan, sehingga Anda tidak dapat mensimulasikan gangguan Availability Zone secara manual.

Meluncurkan domain OpenSearch Layanan Amazon Anda dalam VPC

Anda dapat meluncurkan AWS sumber daya, seperti domain OpenSearch Layanan Amazon, ke cloud pribadi virtual (VPC). VPC adalah jaringan virtual yang didedikasikan untuk Anda. Akun AWSVPC diisolasi secara logis dari jaringan virtual lain di AWS Cloud. Menempatkan domain OpenSearch Layanan dalam VPC memungkinkan komunikasi yang aman antara OpenSearch

Layanan dan layanan lain dalam VPC tanpa memerlukan gateway internet, perangkat NAT, atau koneksi VPN. Semua lalu lintas tetap aman di dalam AWS Cloud.

> **Note**
>
> Jika Anda menempatkan domain OpenSearch Layanan Anda dalam VPC, komputer Anda harus dapat terhubung ke VPC. Sambungan ini sering berupa VPN, transit gateway, jaringan terkelola, atau server proksi. Anda tidak dapat langsung mengakses domain Anda dari luar VPC.

Topik

VPC versus domain publik

Berikut ini adalah beberapa cara domain VPC berbeda dari domain publik. Setiap perbedaan dijelaskan nanti secara lebih rinci.

- Karena isolasi logisnya, domain yang berada di dalam VPC memiliki lapisan keamanan ekstra dibandingkan dengan domain yang menggunakan titik akhir publik.
- Meskipun domain publik dapat diakses dari perangkat apa pun yang terhubung ke internet, domain VPC memerlukan beberapa bentuk VPN atau proxy.
- Dibandingkan dengan domain publik, domain VPC menampilkan lebih sedikit informasi di konsol. Secara khusus, tab kesehatan Cluster tidak menyertakan informasi pecahan, dan tab Indeks tidak ada.
- Titik akhir domain mengambil bentuk yang berbeda (`https://search-domain-namevs.https://vpc-domain-name`).
- Anda tidak dapat menerapkan kebijakan akses berbasis IP ke domain yang berada dalam VPC karena grup keamanan sudah menerapkan kebijakan akses berbasis IP.

Batasan

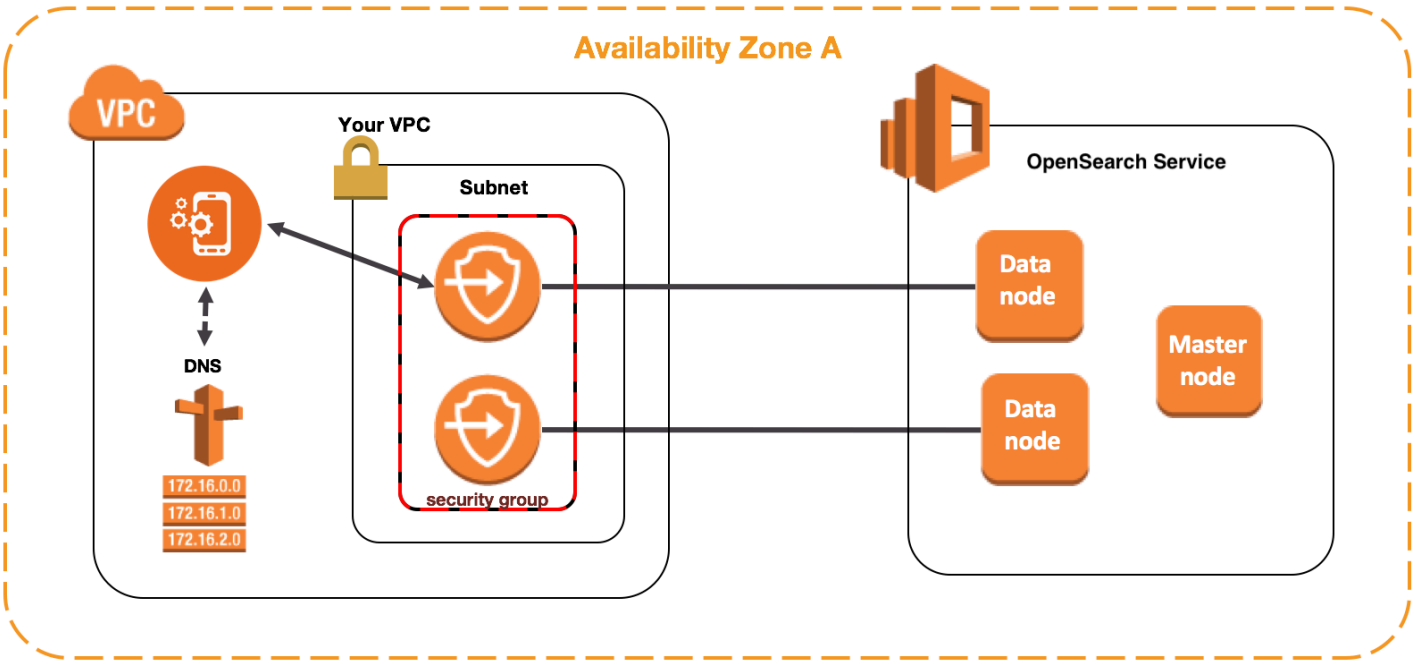
Mengoperasikan domain OpenSearch Layanan dalam VPC memiliki batasan sebagai berikut:

- Jika Anda meluncurkan domain baru dalam VPC, Anda tidak dapat kemudian beralih untuk menggunakan titik akhir publik. Kebalikannya juga benar: Jika Anda membuat domain dengan titik akhir publik, Anda tidak dapat kemudian menempatkannya dalam VPC. Sebagai gantinya, Anda harus membuat domain baru dan memigrasi data Anda.
- Anda dapat meluncurkan domain Anda dalam VPC atau menggunakan titik akhir publik, tetapi Anda tidak dapat melakukan keduanya. Anda harus memilih satu atau yang lain saat membuat domain.
- Anda tidak dapat meluncurkan domain Anda dalam VPC yang menggunakan penyewaan terdedikasi. Anda harus menggunakan VPC dengan penyewaan diatur ke Default.
- Setelah Anda menempatkan domain dalam VPC, Anda tidak dapat memindahkannya ke VPC lain, tetapi Anda dapat mengubah subnet dan pengaturan grup keamanan.
- Untuk mengakses instalasi default OpenSearch Dasbor untuk domain yang berada dalam VPC, pengguna harus memiliki akses ke VPC. Proses ini bervariasi menurut konfigurasi jaringan, tetapi mungkin melibatkan koneksi ke VPN atau jaringan terkelola atau menggunakan server proksi atau transit gateway. Untuk mempelajari selengkapnya, lihat the section called “Tentang kebijakan akses pada domain VPC”, Panduan Pengguna Amazon VPC, dan the section called “Mengontrol akses ke OpenSearch Dasbor”.

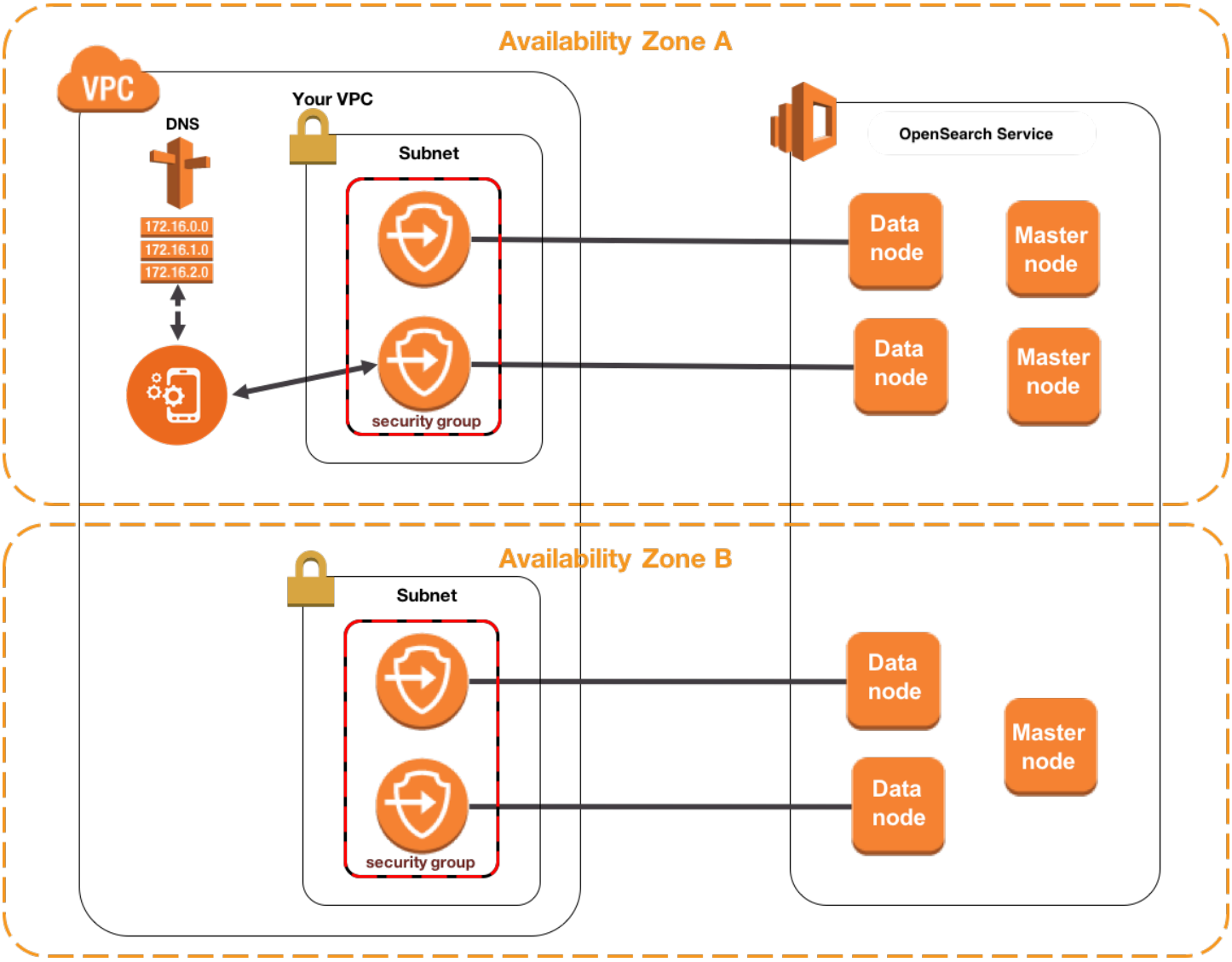
Arsitektur

Untuk mendukung VPC, OpenSearch Service menempatkan titik akhir menjadi satu, dua, atau tiga subnet VPC Anda. Jika Anda mengaktifkan beberapa Availability Zone untuk domain Anda, setiap subnet harus berada di Availability Zone yang berbeda di wilayah yang sama. Jika Anda hanya menggunakan satu Availability Zone, OpenSearch Service menempatkan endpoint ke hanya satu subnet.

Ilustrasi berikut menunjukkan arsitektur VPC untuk satu Availability Zone:



Ilustrasi berikut menunjukkan arsitektur VPC untuk dua Availability Zone:



OpenSearch Layanan juga menempatkan elastic network interface (ENI) di VPC untuk setiap node data Anda. OpenSearch Layanan memberikan setiap ENI alamat IP pribadi dari rentang alamat IPv4 subnet Anda. Layanan ini juga memberikan nama host DNS publik (yang merupakan titik akhir domain) untuk alamat IP. Anda harus menggunakan layanan DNS publik untuk menyelesaikan titik akhir (yang merupakan nama host DNS) ke alamat IP yang sesuai untuk simpul data:

- Jika VPC Anda menggunakan server DNS yang disediakan Amazon dengan menyetel `enableDnsSupport` opsi ke `true` (nilai default), resolusi untuk titik akhir Layanan akan OpenSearch berhasil.
- Jika VPC Anda menggunakan server DNS pribadi dan server dapat menjangkau server DNS otoritatif publik untuk menyelesaikan nama host DNS, resolusi untuk titik akhir Layanan juga akan berhasil. OpenSearch

Karena alamat IP mungkin berubah, Anda harus menyelesaikan titik akhir domain secara berkala sehingga Anda selalu dapat mengakses simpul data yang benar. Kami menyarankan Anda mengatur interval resolusi DNS ke satu menit. Jika Anda menggunakan klien, Anda juga harus memastikan bahwa cache DNS di klien dihapus.

Migrasi dari akses publik ke akses VPC

Ketika Anda membuat domain, Anda menentukan apakah itu harus memiliki titik akhir publik atau berada dalam VPC. Setelah dibuat, Anda tidak dapat beralih dari satu ke yang lain. Sebagai gantinya, Anda harus membuat domain baru dan mengindeks ulang atau memigrasi data Anda secara manual. Snapshots menawarkan cara yang mudah untuk memigrasi data. Untuk informasi tentang mengambil dan memulihkan snapshots, lihat the section called “Membuat snapshot indeks”.

Tentang kebijakan akses pada domain VPC

Menempatkan domain OpenSearch Layanan Anda dalam VPC memberikan lapisan keamanan yang melekat dan kuat. Ketika Anda membuat domain dengan akses publik, titik akhir mengambil bentuk berikut:

```
https://search-domain-name-identifier.region.es.amazonaws.com
```

Seperti yang disarankan oleh label “publik”, titik akhir ini dapat diakses dari perangkat apa pun yang terhubung ke internet, meskipun Anda dapat (dan seharusnya) mengontrol akses ke sana. Jika Anda mengakses titik akhir di peramban web, Anda mungkin menerima pesan `Not Authorized`, tetapi permintaan mencapai domain.

Saat Anda membuat domain dengan akses VPC, titik akhirnya terlihat mirip dengan titik akhir publik:

```
https://vpc-domain-name-identifier.region.es.amazonaws.com
```

Namun, jika Anda mencoba mengakses titik akhir di peramban web, Anda mungkin mendapati bahwa waktu permintaan habis. Untuk melakukan bahkan permintaan `GET` dasar, komputer Anda harus dapat terhubung ke VPC. Sambungan ini sering berupa VPN, transit gateway, jaringan terkelola, atau server proksi. Untuk detail tentang berbagai formulir yang dapat diambil, lihat Contoh untuk VPC di Panduan Pengguna Amazon VPC. Untuk contoh yang terfokus pada pengembangan, lihat the section called “Menguji domain VPC”.

Selain persyaratan konektivitas ini, VPC memungkinkan Anda mengelola akses ke domain melalui grup keamanan. Untuk banyak kasus penggunaan, kombinasi fitur keamanan ini sudah cukup, dan Anda mungkin merasa nyaman menerapkan kebijakan akses terbuka ke domain.

Beroperasi dengan kebijakan akses terbuka tidak berarti bahwa siapa pun di internet dapat mengakses domain OpenSearch Layanan. Sebaliknya, ini berarti bahwa jika permintaan mencapai domain OpenSearch Layanan dan grup keamanan terkait mengizinkannya, domain menerima permintaan tersebut. Satu-satunya pengecualian adalah jika Anda menggunakan kontrol akses berbutir halus atau kebijakan akses yang menentukan peran IAM. Dalam situasi ini, agar domain menerima permintaan, grup keamanan harus mengizinkannya dan itu harus ditandatangani dengan kredensial yang valid.

> **Note**
>
> Karena grup keamanan sudah menerapkan kebijakan akses berbasis IP, Anda tidak dapat menerapkan kebijakan akses berbasis IP ke domain OpenSearch Layanan yang berada dalam VPC. Jika Anda menggunakan akses publik, kebijakan berbasis IP masih tersedia.

Sebelum Anda memulai: prasyarat untuk akses VPC

Sebelum Anda dapat mengaktifkan koneksi antara VPC dan domain OpenSearch Layanan baru Anda, Anda harus melakukan hal berikut:

- Buat VPC

  Untuk membuat VPC, Anda dapat menggunakan konsol VPC Amazon, AWS CLI, atau salah satu SDK. AWS Untuk informasi lebih lanjut, lihat Bekerja dengan VPC di Panduan Pengguna Amazon VPC. Jika Anda sudah memiliki VPC, Anda dapat melewati langkah ini.

- Cadangan alamat IP

  OpenSearch Layanan memungkinkan koneksi VPC ke domain dengan menempatkan antarmuka jaringan di subnet VPC. Setiap antarmuka jaringan dikaitkan dengan alamat IP. Anda harus menyimpan alamat IP dalam jumlah yang cukup di subnet untuk antarmuka jaringan. Untuk informasi selengkapnya, lihat Menyimpan alamat IP di subnet VPC.

Menguji domain VPC

Keamanan VPC yang ditingkatkan dapat membuat koneksi ke domain Anda dan menjalankan tes dasar menjadi tantangan. Jika Anda sudah memiliki domain VPC OpenSearch Layanan dan lebih suka tidak membuat server VPN, coba proses berikut:

1. Untuk kebijakan akses domain Anda, pilih Hanya gunakan kontrol akses berbutir halus. Anda selalu dapat memperbarui pengaturan ini setelah Anda menyelesaikan pengujian.
2. Buat instans Amazon EC2 Amazon Linux di VPC, subnet, dan grup keamanan yang sama dengan domain Layanan Anda. OpenSearch

   Karena instans ini adalah untuk tujuan pengujian dan perlu melakukan sedikit pekerjaan, pilih tipe instans yang murah seperti `t2.micro`. Tetapkan alamat IP publik instans dan buat pasangan kunci baru atau pilih yang sudah ada. Jika Anda membuat kunci baru, unduh ke direktori `~/.ssh` Anda.

   Untuk mempelajari lebih lanjut tentang membuat instans, lihat Memulai instans Amazon EC2 Linux.
3. Tambahkan sebuah gateway internet ke VPC Anda.
4. Di tabel rute untuk VPC Anda, tambahkan rute baru. Untuk Tujuan, tentukan sebuah Blok CIDR yang berisi alamat IP publik komputer Anda. Untuk Target, tentukan gateway internet yang baru saja Anda buat.

   Misalnya, Anda dapat menentukan `123.123.123.123/32` hanya untuk komputer Anda atau `123.123.123.0/24` untuk berbagai komputer.
5. Untuk grup keamanan, tentukan dua aturan masuk:

Tipe	Protokol	Baris Port	Sumber
SSH (22)	TCP (6)	22	*your-cidr-block*
HTTPS (443)	TCP (6)	443	*your-security-group-id*

   Aturan pertama memungkinkan Anda SSH ke instans EC2 Anda. Yang kedua memungkinkan instans EC2 untuk berkomunikasi dengan domain OpenSearch Layanan melalui HTTPS.
6. Dari terminal, jalankan perintah berikut:

```
ssh -i ~/.ssh/your-key.pem ec2-user@your-ec2-instance-public-ip -N -L
 9200:vpc-domain-name.region.es.amazonaws.com:443
```

Perintah ini membuat terowongan SSH yang meneruskan permintaan ke https://localhost:9200 ke domain OpenSearch Layanan Anda melalui instans EC2. Menentukan port 9200 dalam perintah mensimulasikan OpenSearch instalasi lokal, tetapi gunakan port mana pun yang Anda inginkan. OpenSearch Layanan hanya menerima koneksi melalui port 80 (HTTP) atau 443 (HTTPS).

Perintah tidak memberikan umpan balik dan berjalan tanpa batas waktu. Untuk menghentikannya, tekan `Ctrl + C`.

7. Arahkan ke https://localhost:9200/_dashboards/ di browser web Anda. Anda mungkin perlu menyatakan pengecualian keamanan.

   Sebagai alternatif, Anda dapat mengirim permintaan ke https://localhost:9200 menggunakan curl, Postman, atau bahasa pemrograman favorit Anda.

   > **Tip**
   >
   > Jika Anda mengalami kesalahan curl karena ketidakcocokan sertifikat, coba flag `--insecure`.

Menyimpan alamat IP di subnet VPC

OpenSearch Layanan menghubungkan domain ke VPC dengan menempatkan antarmuka jaringan di subnet VPC (atau beberapa subnet VPC jika Anda mengaktifkan beberapa Availability Zone). Setiap antarmuka jaringan dikaitkan dengan alamat IP. Sebelum Anda membuat domain OpenSearch Layanan Anda, Anda harus memiliki cukup banyak alamat IP yang tersedia di setiap subnet untuk mengakomodasi antarmuka jaringan.

Berikut rumus dasarnya: Jumlah alamat IP yang dicadangkan OpenSearch Layanan di setiap subnet adalah tiga kali jumlah node data, dibagi dengan jumlah Availability Zones.

Contoh

- Jika domain memiliki sembilan node data di tiga Availability Zone, jumlah IP per subnet adalah 9 * 3/3 = 9.

- Jika domain memiliki delapan node data di dua Availability Zone, jumlah IP per subnet adalah 8 * 3/2 = 12.
- Jika domain memiliki enam node data dalam satu Availability Zone, jumlah IP per subnet adalah 6 * 3/1 = 18.

Saat Anda membuat domain, OpenSearch Layanan menyimpan alamat IP, menggunakan beberapa untuk domain, dan menyimpan sisanya untuk penerapan biru/hijau. Anda dapat melihat antarmuka jaringan dan alamat IP terkaitnya di bagian Antarmuka Jaringan dari konsol Amazon EC2. Kolom Deskripsi menunjukkan domain OpenSearch Layanan mana yang terkait dengan antarmuka jaringan.

> **Tip**
>
> Kami menyarankan Anda membuat subnet khusus untuk alamat IP yang dicadangkan OpenSearch Layanan. Dengan menggunakan subnet terdedikasi, Anda menghindari tumpang tindih dengan aplikasi dan layanan lain dan memastikan bahwa Anda dapat menyimpan alamat IP tambahan jika Anda perlu untuk menskalakan klaster Anda di masa mendatang. Untuk mempelajari selengkapnya, lihat Membuat subnet di VPC Anda.

Peran yang terhubung dengan layanan untuk akses VPC

Peran terkait layanan adalah jenis peran IAM unik yang mendelegasikan izin ke layanan sehingga dapat membuat dan mengelola sumber daya atas nama Anda. OpenSearch Layanan memerlukan peran terkait layanan untuk mengakses VPC Anda, membuat titik akhir domain, dan menempatkan antarmuka jaringan di subnet VPC Anda.

OpenSearch Layanan secara otomatis membuat peran saat Anda menggunakan konsol OpenSearch Layanan untuk membuat domain dalam VPC. Agar pembuatan otomatis ini berhasil, Anda harus memiliki izin untuk `iam:CreateServiceLinkedRole` tindakan tersebut. Untuk mempelajari selengkapnya, lihat Izin peran terkait layanan di Panduan Pengguna IAM.

Setelah OpenSearch Layanan membuat peran, Anda dapat melihatnya (`AWSServiceRoleForAmazonOpenSearchService`) menggunakan konsol IAM.

Untuk informasi lengkap tentang izin peran ini dan cara menghapusnya, lihat the section called “Menggunakan peran terkait layanan”.

Membuat snapshot indeks di Amazon Service OpenSearch

Snapshot di Amazon OpenSearch Service adalah cadangan indeks dan status klaster. Status termasuk pengaturan klaster, informasi simpul, pengaturan indeks, dan alokasi serpihan.

OpenSearch Snapshot layanan datang dalam bentuk berikut:

- Snapshot otomatis hanya untuk pemulihan klaster. Anda dapat menggunakannya untuk memulihkan domain Anda dalam peristiwa status klaster merah atau kehilangan data. Untuk informasi selengkapnya, lihat Memulihkan snapshot di bawah ini. OpenSearch Layanan menyimpan snapshot otomatis dalam bucket Amazon S3 yang telah dikonfigurasi sebelumnya tanpa biaya tambahan.
- Snapshot manual adalah untuk pemulihan klaster atau untuk memindahkan data dari satu klaster ke klaster lainnya. Anda harus memulai snapshot manual. Snapshot ini disimpan dalam bucket Amazon S3 Anda sendiri dan biaya S3 standar berlaku. Jika Anda memiliki snapshot dari OpenSearch kluster yang dikelola sendiri, Anda dapat menggunakan snapshot tersebut untuk bermigrasi ke domain Layanan. OpenSearch Untuk informasi selengkapnya, lihat Migrasi ke OpenSearch Layanan Amazon.

Semua domain OpenSearch Layanan mengambil snapshot otomatis, tetapi frekuensinya berbeda dengan cara berikut:

- Untuk domain yang berjalan OpenSearch atau Elasticsearch 5.3 dan yang lebih baru, OpenSearch Layanan mengambil snapshot otomatis setiap jam dan mempertahankan hingga 336 di antaranya selama 14 hari. Cuplikan per jam tidak terlalu mengganggu karena sifat inkrementalnya. Mereka juga menyediakan titik pemulihan yang lebih baru jika terjadi masalah domain.
- Untuk domain yang menjalankan Elasticsearch 5.1 dan yang lebih lama, OpenSearch Layanan mengambil snapshot otomatis harian selama jam yang Anda tentukan, mempertahankan hingga 14 di antaranya, dan tidak menyimpan data snapshot apa pun selama lebih dari 30 hari.

Jika klaster Anda memasuki status merah, semua snapshot otomatis gagal selagi status klaster tetap ada. Jika Anda tidak memperbaiki masalah dalam waktu dua minggu, Anda dapat kehilangan data di klaster Anda secara permanen. Untuk langkah-langkah pemecahan masalah, lihat the section called “Status klaster merah”.

Topik

- Prasyarat

- Mendaftarkan repositori snapshot manual
- Mengambil snapshot manual
- Memulihkan snapshot
- Menghapus snapshot manual
- Mengotomatiskan snapshot dengan Manajemen Snapshot
- Mengotomatisasi snapshot dengan Manajemen State Indeks
- Menggunakan Curator untuk snapshot

Prasyarat

Untuk membuat snapshot secara manual, Anda perlu bekerja dengan IAM dan Amazon S3. Pastikan Anda memenuhi prasyarat berikut sebelum Anda mencoba untuk mengambil snapshot:

Prasyarat	Deskripsi
Bucket S3	Buat bucket S3 untuk menyimpan snapshot manual untuk domain OpenSearch Layanan Anda. Untuk petunjuknya, lihat Membuat Bucket di Panduan Pengguna Layanan Penyimpanan Sederhana Amazon. Ingat nama bucket untuk menggunakannya di tempat-tempat berikut: • Pernyataan `Resource` dari kebijakan IAM yang dilampirkan pada IAM role Anda • Klien Python digunakan untuk mendaftarkan repositori snapshot (jika Anda menggunakan metode ini) **Important** Jangan menerapkan aturan siklus hidup S3 Glacier ke bucket ini. Snapshot manual tidak mendukung kelas penyimpanan S3 Glacier.
Peran IAM	Buat peran IAM untuk mendelegasikan izin ke Layanan. OpenSearch Untuk petunjuk, lihat Membuat IAM role (konsol) di Panduan Pengguna IAM. Sisa dari bab ini mengacu pada peran ini sebagai `TheSnapshotRole` . Lampirkan kebijakan IAM

Prasyarat	Deskripsi
	Lampirkan kebijakan berikut ke `TheSnapshotRole` untuk mengizinkan akses ke bucket S3:

```
{
  "Version": "2012-10-17",
  "Statement": [{
      "Action": [
        "s3:ListBucket"
      ],
      "Effect": "Allow",
      "Resource": [
        "arn:aws:s3::: s3-bucket-name "
      ]
    },
    {
      "Action": [
        "s3:GetObject",
        "s3:PutObject",
        "s3:DeleteObject"
      ],
      "Effect": "Allow",
      "Resource": [
        "arn:aws:s3::: s3-bucket-name /*"
      ]
    }
  ]
}
```

Untuk petunjuk guna melampirkan kebijakan ke peran, lihat Menambahkan Izin Identitas IAM di Panduan Pengguna IAM.

Edit hubungan kepercayaan

Edit hubungan kepercayaan `TheSnapshotRole` untuk menentukan OpenSearch Layanan dalam `Principal` pernyataan seperti yang ditunjukkan pada contoh berikut:

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

Prasyarat	Deskripsi

	<pre>"Sid": "",
	"Effect": "Allow",
	"Principal": {
	"Service": "es.amazonaws.com"
	},
	"Action": "sts:AssumeRole"
	}]
	
	}</pre>
	Untuk petunjuk guna mengedit hubungan kepercayaan, lihat Mengubah kebijakan kepercayaan peran di Panduan Pengguna IAM.

Prasyarat	Deskripsi

Izin	Untuk mendaftarkan repositori snapshot, Anda harus dapat meneruskan `TheSnapshotRole` ke Layanan. OpenSearch Anda juga memerlukan akses ke tindakan `es:ESHttpPut` . Untuk memberikan kedua izin ini, lampirkan kebijakan berikut ke peran IAM yang kredensialnya digunakan untuk menandatangani permintaan:
	
	`{ "Version": "2012-10-17", "Statement": [ { "Effect": "Allow", "Action": "iam:PassRole", "Resource": "arn:aws:iam:: 123456789012 :role/TheSnapshotRole " }, { "Effect": "Allow", "Action": "es:ESHttpPut", "Resource": "arn:aws:es: region:123456789012 :domain/domain-name /*" } ] }`
	
	Jika pengguna atau peran Anda tidak memiliki `iam:PassRole` izin untuk diteruskan`TheSnapshotRole` , Anda mungkin mengalami kesalahan umum berikut saat mencoba mendaftarkan repositori di langkah berikutnya:
	
	`$ python register-repo.py {"Message":"User: arn:aws:iam:: 123456789012 :user/MyUserAccount is not authorized to perform: iam:PassRole on resource: arn:aws:iam:: 123456789012 :role/TheSnapshotRole "}`

Mendaftarkan repositori snapshot manual

Anda perlu mendaftarkan repositori snapshot dengan OpenSearch Service sebelum Anda dapat mengambil snapshot indeks manual. Operasi satu kali ini mengharuskan Anda menandatangani

AWS permintaan dengan kredensil yang diizinkan untuk diaksesTheSnapshotRole, seperti yang dijelaskan dalam. the section called “Prasyarat”

Langkah 1: Petakan peran snapshot di OpenSearch Dasbor (jika menggunakan kontrol akses berbutir halus)

Kontrol akses detail memperkenalkan langkah tambahan saat mendaftarkan repositori. Bahkan jika Anda menggunakan autentikasi basic HTTP untuk semua tujuan lain, Anda perlu memetakan peran `manage_snapshots` ke IAM role Anda yang memiliki izin `iam:PassRole` untuk meneruskan `TheSnapshotRole`.

1. Arahkan ke plugin OpenSearch Dasbor untuk domain OpenSearch Layanan Anda. Anda dapat menemukan titik akhir Dasbor di dasbor domain Anda di konsol OpenSearch Layanan.
2. Dari menu utama, pilih Keamanan, Peran, lalu pilih peran manage_snapshots.
3. Pilih Pengguna yang Dipetakan, Kelola pemetaan.
4. Tambahkan ARN dari peran yang memiliki izin untuk diteruskan. `TheSnapshotRole` Letakkan ARN peran di bawah peran Backend.

   ```
   arn:aws:iam::123456789123:role/role-name
   ```

5. Pilih Peta dan konfirmasikan pengguna atau peran muncul di bawah Pengguna yang dipetakan.

Langkah 2: Mendaftarkan repositori

Tab Snapshots berikut menunjukkan cara mendaftarkan direktori snapshot. Untuk opsi khusus untuk mengenkripsi snapshot manual dan mendaftarkan snapshot setelah bermigrasi ke domain baru, lihat tab yang relevan.

Snapshots

Untuk mendaftarkan repositori snapshot, kirim permintaan PUT ke titik akhir domain OpenSearch Layanan. Anda dapat menggunakan curl, contoh klien Python, Postman, atau metode lain untuk mengirim permintaan yang ditandatangani untuk mendaftarkan repositori snapshot. Perhatikan bahwa Anda tidak dapat menggunakan permintaan PUT di konsol OpenSearch Dasbor untuk mendaftarkan repositori.

Permintaan mengambil format berikut:

```
PUT domain-endpoint/_snapshot/my-snapshot-repo-name
```

```
{
  "type": "s3",
  "settings": {
    "bucket": "s3-bucket-name",
    "base_path": "my/snapshot/directory",
    "region": "region",
    "role_arn": "arn:aws:iam::123456789012:role/TheSnapshotRole"
  }
}
```

> **Note**
>
> Nama repositori tidak dapat dimulai dengan “cs-”. Selain itu, Anda tidak boleh menulis ke repositori yang sama dari beberapa domain. Hanya satu domain yang harus memiliki akses tulis ke repositori.

Jika domain Anda berada dalam virtual private cloud (VPC), komputer Anda harus terhubung ke VPC agar permintaan berhasil mendaftarkan repositori snapshot. Mengakses VPC bervariasi menurut konfigurasi jaringan, tetapi kemungkinan melibatkan koneksi ke VPN atau jaringan perusahaan. Untuk memeriksa apakah Anda dapat mencapai domain OpenSearch Layanan, `https://your-vpc-domain.region.es.amazonaws.com` navigasikan ke browser web dan verifikasi bahwa Anda menerima respons JSON default.

Jika bucket Amazon S3 Anda berada di tempat lain Wilayah AWS selain OpenSearch domain Anda, tambahkan parameter `"endpoint": "s3.amazonaws.com"` ke permintaan.

Encrypted snapshots

Saat ini Anda tidak dapat menggunakan kunci AWS Key Management Service (KMS) untuk mengenkripsi snapshot manual, tetapi Anda dapat melindunginya menggunakan enkripsi sisi server (SSE).

Untuk mengaktifkan SSE dengan kunci terkelola S3 untuk bucket yang Anda gunakan sebagai repositori snapshot, tambahkan `"server_side_encryption": true` ke blok permintaan PUT. `"settings"` Untuk informasi selengkapnya, lihat Melindungi data menggunakan enkripsi sisi server dengan kunci enkripsi terkelola Amazon S3 di Panduan Pengguna Amazon Simple Storage Service.

Atau, Anda dapat menggunakan AWS KMS kunci untuk enkripsi sisi server pada bucket S3 yang Anda gunakan sebagai repositori snapshot. Jika Anda menggunakan pendekatan ini,

pastikan untuk memberikan `TheSnapshotRole` izin ke AWS KMS kunci yang digunakan untuk mengenkripsi bucket S3. Untuk informasi selengkapnya, lihat Kebijakan utama di AWS KMS.

Domain migration

Mendaftarkan repositori snapshot adalah operasi satu kali. Namun, untuk bermigrasi dari satu domain ke domain lain, Anda harus mendaftarkan repositori snapshot yang sama pada domain lama dan domain baru. Nama repositori adalah semaunya.

Pertimbangkan panduan berikut saat bermigrasi ke domain baru atau mendaftarkan repositori yang sama dengan beberapa domain:

- Saat mendaftarkan repositori pada domain baru, tambahkan `"readonly": true` ke blok `"settings"` dari permintaan PUT. Pengaturan ini mencegah Anda dari secara tidak sengaja menimpa data dari domain lama. Hanya satu domain yang harus memiliki akses tulis ke repositori.
- Jika Anda memigrasikan data ke domain yang berbeda Wilayah AWS, (misalnya, dari domain lama dan bucket yang terletak di us-east-2 ke domain baru di us-west-2), ganti `"region": "region"` dengan `"endpoint": "s3.amazonaws.com"` di pernyataan PUT dan coba lagi permintaan tersebut.

Menggunakan sampel klien Python

Klien Python lebih mudah untuk diotomatisasi daripada permintaan HTTP sederhana dan memiliki kemampuan penggunaan kembali yang lebih baik. Jika anda memilih untuk menggunakan metode ini untuk mendaftarkan repositori snapshot, simpan sampel kode Python berikut sebagai file Python, seperti `register-repo.py`. Klien memerlukan paket AWS SDK for Python (Boto3), request dan requests-aws4auth. Klien berisi contoh-contoh yang dikomentari untuk operasi snapshot lainnya.

Memperbarui variabel berikut dalam kode sampel: `host`, `region`, `path`, dan `payload`.

```
import boto3
import requests
from requests_aws4auth import AWS4Auth

host = '' # domain endpoint
region = '' # e.g. us-west-1
service = 'es'
credentials = boto3.Session().get_credentials()
awsauth = AWS4Auth(credentials.access_key, credentials.secret_key, region, service,
 session_token=credentials.token)
```

```
# Register repository

path = '/_snapshot/my-snapshot-repo-name' # the OpenSearch API endpoint
url = host + path

payload = {
  "type": "s3",
  "settings": {
    "bucket": "s3-bucket-name",
    "base_path": "my/snapshot/directory",
    "region": "us-west-1",
    "role_arn": "arn:aws:iam::123456789012:role/snapshot-role"
  }
}

headers = {"Content-Type": "application/json"}

r = requests.put(url, auth=awsauth, json=payload, headers=headers)

print(r.status_code)
print(r.text)

# # Take snapshot
#
# path = '_snapshot/my-snapshot-repo-name/my-snapshot'
# url = host + path
#
# r = requests.put(url, auth=awsauth)
#
# print(r.text)
#
# # Delete index
#
# path = 'my-index'
# url = host + path
#
# r = requests.delete(url, auth=awsauth)
#
# print(r.text)
#
# # Restore snapshot (all indexes except Dashboards and fine-grained access control)
#
# path = '_snapshot/my-snapshot-repo-name/my-snapshot/_restore'
```

```
# url = host + path
#
# payload = {
#   "indices": "-.kibana*,-.opendistro_security,-.opendistro-*",
#   "include_global_state": False
# }
#
# headers = {"Content-Type": "application/json"}
#
# r = requests.post(url, auth=awsauth, json=payload, headers=headers)
#
# print(r.text)
#
# # Restore snapshot (one index)
#
# path = '_snapshot/my-snapshot-repo-name/my-snapshot/_restore'
# url = host + path
#
# payload = {"indices": "my-index"}
#
# headers = {"Content-Type": "application/json"}
#
# r = requests.post(url, auth=awsauth, json=payload, headers=headers)
#
# print(r.text)
```

Mengambil snapshot manual

Snapshot tidak seketika. Mereka membutuhkan waktu untuk menyelesaikan dan tidak mewakili point-in-time pandangan cluster yang sempurna. Sementara snapshot sedang berlangsung, Anda masih dapat mengindeks dokumen dan membuat permintaan lain ke klaster, tetapi dokumen baru dan pembaruan untuk dokumen yang ada umumnya tidak termasuk dalam snapshot. Snapshot menyertakan pecahan primer seperti yang ada saat OpenSearch memulai snapshot. Tergantung pada ukuran kolam utas snapshot Anda, serpihan yang berbeda mungkin disertakan dalam snapshot pada waktu yang sedikit berbeda. Untuk praktik terbaik snapshot, lihatthe section called “Tingkatkan kinerja snapshot”.

Penyimpanan dan performa snapshot

OpenSearch snapshot bersifat inkremental, artinya mereka hanya menyimpan data yang berubah sejak snapshot terakhir yang berhasil. Sifat pertambahan ini berarti perbedaan dalam penggunaan

disk antara snapshot yang sering dan jarang sering kali minimal. Dengan kata lain, mengambil snapshot per jam selama seminggu (untuk total 168 snapshot) mungkin tidak menggunakan lebih banyak ruang disk daripada mengambil satu snapshot pada akhir minggu. Selain itu, semakin sering Anda mengambil snapshot, semakin sedikit waktu yang diperlukan untuk menyelesaikannya. Misalnya, snapshot harian dapat memakan waktu 20-30 menit untuk diselesaikan, sedangkan snapshot per jam mungkin selesai dalam beberapa menit. Beberapa OpenSearch pengguna mengambil foto sesering setiap setengah jam.

Ambil snapshot

Anda menentukan informasi berikut saat Anda membuat snapshot:

- Nama dari repositori snapshot Anda
- Sebuah nama untuk snapshot

Contoh dalam bab ini menggunakan curl, klien HTTP umum, untuk kenyamanan dan singkatnya. Untuk meneruskan nama pengguna dan kata sandi ke permintaan curl Anda, lihat tutorial Memulai.

Jika kebijakan akses menentukan pengguna atau peran, Anda harus menandatangani permintaan snapshot. Untuk curl, Anda dapat menggunakan `--aws-sigv4opsi` dengan versi 7.75.0 atau yang lebih baru. Anda juga dapat menggunakan contoh yang dikomentari dalam contoh klien Python untuk membuat permintaan HTTP yang ditandatangani ke titik akhir yang sama dengan yang digunakan perintah curl.

Untuk mengambil snapshot manual, lakukan langkah-langkah berikut:

1. Anda tidak dapat mengambil snapshot jika salah satu sedang berlangsung. Untuk memeriksa, jalankan perintah berikut:

   ```
   curl -XGET 'domain-endpoint/_snapshot/_status'
   ```

2. Jalankan perintah berikut untuk mengambil snapshot manual:

   ```
   curl -XPUT 'domain-endpoint/_snapshot/repository-name/snapshot-name'
   ```

   Untuk menyertakan atau mengecualikan indeks tertentu dan menentukan pengaturan lain, tambahkan isi permintaan. Untuk struktur permintaan, lihat Mengambil snapshot dalam OpenSearch dokumentasi.

> **Note**
>
> Waktu yang diperlukan untuk mengambil snapshot meningkat dengan ukuran domain OpenSearch Layanan. Operasi snapshot yang berjalan lama terkadang mengalami kesalahan berikut: `504 GATEWAY_TIMEOUT`. Anda biasanya dapat mengabaikan kesalahan ini dan menunggu operasi selesai dengan sukses. Jalankan perintah berikut untuk memverifikasi status dari semua snapshot domain Anda:
>
> ```
> curl -XGET 'domain-endpoint/_snapshot/repository-name/_all?pretty'
> ```

Memulihkan snapshot

Sebelum mengembalikan snapshot, pastikan domain tujuan tidak menggunakan Multi-AZ dengan Standby. Mengaktifkan siaga menyebabkan operasi pemulihan gagal.

> **Warning**
>
> Jika Anda menggunakan alias indeks, Anda harus menghentikan permintaan tulis ke alias atau mengalihkan alias ke indeks lain sebelum menghapus indeksnya. Menghentikan permintaan penulisan membantu menghindari skenario berikut:
>
> 1. Anda menghapus indeks, yang juga menghapus aliasnya.
> 2. Permintaan penulisan yang salah ke alias yang sekarang dihapus membuat indeks baru dengan nama yang sama dengan alias.
> 3. Anda tidak dapat lagi menggunakan alias karena konflik penamaan dengan indeks baru. Jika Anda telah mengalihkan alias ke indeks lain, tentukan `"include_aliases": false` saat Anda memulihkan dari snapshot.

Untuk mengembalikan snapshot

1. Identifikasi snapshot yang ingin Anda pulihkan. Pastikan bahwa semua pengaturan untuk indeks ini, seperti paket penganalisis khusus atau pengaturan persyaratan alokasi, kompatibel dengan domain. Untuk melihat semua repositori snapshot, jalankan perintah berikut:

   ```
   curl -XGET 'domain-endpoint/_snapshot?pretty'
   ```

Setelah Anda mengidentifikasi repositori, jalankan perintah berikut untuk melihat semua snapshot:

```
curl -XGET 'domain-endpoint/_snapshot/repository-name/_all?pretty'
```

> **Note**
>
> Sebagian besar snapshot otomatis disimpan dalam repositori `cs-automated`. Jika domain Anda mengenkripsi data saat tidak aktif, maka data tersebut akan disimpan di repositori `cs-automated-enc`. Jika Anda tidak melihat repositori snapshot manual yang Anda cari, pastikan Anda mendaftarkannya ke domain.

2. (Opsional) Hapus atau ganti nama satu atau beberapa indeks di domain OpenSearch Layanan jika Anda memiliki konflik penamaan antara indeks di cluster dan indeks dalam snapshot. Anda tidak dapat mengembalikan snapshot indeks Anda ke OpenSearch cluster yang sudah berisi indeks dengan nama yang sama.

   Anda memiliki opsi berikut jika Anda memiliki konflik penamaan indeks:

   - Hapus indeks pada domain OpenSearch Layanan yang ada dan kemudian pulihkan snapshot.
   - Ganti nama indeks saat Anda memulihkannya dari snapshot dan indeks ulang nanti.
   - Kembalikan snapshot ke domain OpenSearch Layanan yang berbeda (hanya mungkin dengan snapshot manual).

   Perintah berikut menghapus semua indeks yang ada di domain:

   ```
   curl -XDELETE 'domain-endpoint/_all'
   ```

   Namun, jika Anda tidak berencana untuk memulihkan semua indeks, Anda cukup menghapus satu:

   ```
   curl -XDELETE 'domain-endpoint/index-name'
   ```

3. Untuk memulihkan snapshot, jalankan perintah berikut:

   ```
   curl -XPOST 'domain-endpoint/_snapshot/repository-name/snapshot-name/_restore'
   ```

Karena izin khusus pada OpenSearch Dasbor dan indeks kontrol akses berbutir halus, upaya untuk memulihkan semua indeks mungkin gagal, terutama jika Anda mencoba memulihkan dari snapshot otomatis. Contoh berikut memulihkan hanya satu indeks, `my-index`, dari `2020-snapshot` di repositori snapshot `cs-automated`:

```
curl -XPOST 'domain-endpoint/_snapshot/cs-automated/2020-snapshot/_restore' \
-d '{"indices": "my-index"}' \
-H 'Content-Type: application/json'
```

Sebagai alternatif, Anda mungkin ingin memulihkan semua indeks kecuali Dasbor dan indeks kontrol akses berbutir halus:

```
curl -XPOST 'domain-endpoint/_snapshot/cs-automated/2020-snapshot/_restore' \
-d '{"indices": "-.kibana*,-.opendistro*"}' \
-H 'Content-Type: application/json'
```

Anda dapat memulihkan snapshot tanpa menghapus datanya dengan menggunakan parameter `rename_pattern` dan`rename_replacement`. Untuk informasi selengkapnya tentang parameter ini, lihat bidang permintaan Restore Snapshot API dan contoh permintaan dalam OpenSearch dokumentasi.

> **Note**
>
> Jika tidak semua pecahan utama tersedia untuk indeks yang terlibat, snapshot mungkin memiliki file. `state` `PARTIAL` Nilai ini menunjukkan bahwa data dari setidaknya satu serpihan tidak berhasil disimpan. Anda masih dapat memulihkan dari sebagian snapshot, tetapi Anda mungkin perlu menggunakan snapshot lama untuk memulihkan indeks yang hilang.

Menghapus snapshot manual

Untuk menghapus snapshot manual, jalankan perintah berikut:

```
DELETE _snapshot/repository-name/snapshot-name
```

Mengotomatiskan snapshot dengan Manajemen Snapshot

Anda dapat menyiapkan kebijakan Manajemen Snapshot (SM) di OpenSearch Dasbor untuk mengotomatiskan pembuatan dan penghapusan snapshot secara berkala. SM dapat mengambil snapshot dari sekelompok indeks, sedangkan Index State Management hanya dapat mengambil satu snapshot per indeks. Untuk menggunakan SM in OpenSearch Service, Anda harus mendaftarkan repositori Amazon S3 Anda sendiri. Untuk petunjuk untuk mendaftarkan repositori Anda, lihat Mendaftarkan repositori snapshot manual.

Sebelum SM, OpenSearch Layanan menawarkan fitur snapshot otomatis gratis yang masih diaktifkan secara default. Fitur ini mengirimkan snapshot ke repositori yang dikelola layanan`cs-*`. Untuk menonaktifkan fitur, hubungi. AWS Support

Untuk informasi selengkapnya tentang fitur SM, lihat Manajemen snapshot dalam OpenSearch dokumentasi.

SM saat ini tidak mendukung pembuatan snapshot pada beberapa jenis indeks. Misalnya, jika Anda mencoba membuat snapshot pada beberapa indeks dengan * dan beberapa indeks berada di tingkat hangat, pembuatan snapshot akan gagal. Jika Anda memerlukan snapshot berisi beberapa jenis indeks, gunakan tindakan snapshot ISM hingga SM mendukung opsi ini.

Konfigurasi izin

Jika Anda memutakhirkan ke 2.5 dari versi domain OpenSearch Layanan sebelumnya, izin keamanan manajemen snapshot mungkin tidak ditentukan pada domain. Pengguna non-admin harus dipetakan ke peran ini untuk menggunakan manajemen snapshot pada domain menggunakan kontrol akses berbutir halus. Untuk membuat peran manajemen snapshot secara manual, lakukan langkah-langkah berikut:

1. Di OpenSearch Dasbor, buka Keamanan dan pilih Izin.
2. Pilih Buat grup tindakan dan konfigurasi grup-grup berikut:

Nama grup	Izin
`snapshot_management_full_access`	• `cluster:admin/opensearch/snapshot_management/*` • `cluster:admin/opensearch/notifications/feature/publish`

Nama grup	Izin
	• `cluster:admin/repository/*` • `cluster:admin/snapshot/*`
`snapshot_management_read_access`	• `cluster:admin/opensearch/snapshot_management/policy/get` • `cluster:admin/opensearch/snapshot_management/policy/search` • `cluster:admin/opensearch/snapshot_management/policy/explain` • `cluster:admin/repository/get` • `cluster:admin/snapshot/get`

3. Pilih Peran dan Buat peran.
4. Beri nama peran snapshot_management_role.
5. Untuk izin Cluster, pilih `snapshot_management_full_access` atau`snapshot_management_read_access`.
6. Pilih Buat.
7. Setelah Anda membuat peran, petakan ke setiap pengguna atau peran backend yang akan mengelola snapshot.

Pertimbangan

Pertimbangkan hal berikut saat Anda mengonfigurasi manajemen snapshot:

- Satu kebijakan diperbolehkan per repositori.
- Hingga 400 snapshot diizinkan untuk satu kebijakan.
- Fitur ini tidak akan berjalan jika domain Anda memiliki status merah, berada di bawah tekanan JVM tinggi (85% atau lebih tinggi), atau memiliki fungsi snapshot macet. Ketika kinerja pengindeksan dan pencarian keseluruhan klaster Anda terpengaruh, SM juga dapat terpengaruh.
- Operasi snapshot hanya dimulai setelah operasi sebelumnya selesai, sehingga tidak ada operasi snapshot bersamaan yang diaktifkan oleh satu kebijakan.
- Beberapa kebijakan dengan jadwal yang sama dapat menyebabkan lonjakan sumber daya. Jika indeks snapshot kebijakan tumpang tindih, operasi snapshot level shard hanya dapat berjalan

secara berurutan, yang dapat menyebabkan masalah kinerja berjenjang. Jika kebijakan berbagi repositori, akan ada lonjakan operasi penulisan ke repositori tersebut.

- Kami menyarankan Anda menjadwalkan otomatisasi operasi snapshot Anda tidak lebih dari sekali per jam, kecuali jika Anda memiliki kasus penggunaan khusus.

Mengotomatisasi snapshot dengan Manajemen State Indeks

Anda dapat menggunakan `snapshot` operasi Index State Management (ISM) untuk secara otomatis memicu snapshot indeks berdasarkan perubahan usia, ukuran, atau jumlah dokumen. ISM adalah yang terbaik ketika Anda membutuhkan satu snapshot per indeks. Jika Anda perlu memotret sekelompok indeks, lihat. Mengotomatiskan snapshot dengan Manajemen Snapshot

Untuk menggunakan SM in OpenSearch Service, Anda harus mendaftarkan repositori Amazon S3 Anda sendiri. Untuk contoh kebijakan ISM yang menggunakan operasi `snapshot`, lihat Sampel Kebijakan.

Menggunakan Curator untuk snapshot

Jika ISM tidak berfungsi untuk manajemen indeks dan snapshot, Anda dapat menggunakan Kurator sebagai gantinya. Ini menawarkan fungsionalitas penyaringan lanjutan yang dapat membantu menyederhanakan tugas manajemen pada cluster yang kompleks. Gunakan pip untuk menginstal Curator:

```
pip install elasticsearch-curator
```

Anda dapat menggunakan Curator sebagai antarmuka baris perintah (CLI) atau API Python. Jika Anda menggunakan Python API, Anda harus menggunakan versi 7.13.4 atau sebelumnya dari klien elasticsearch-py lama. Itu tidak mendukung klien opensearch-py.

Jika Anda menggunakan CLI, ekspor kredensial Anda pada baris perintah dan konfigurasikan `curator.yml` sebagai berikut:

```
client:
  hosts: search-my-domain.us-west-1.es.amazonaws.com
  port: 443
  use_ssl: True
  aws_region: us-west-1
  aws_sign_request: True
```

```
  ssl_no_validate: False
  timeout: 60

logging:
  loglevel: INFO
```

Memutakhirkan domain OpenSearch Layanan Amazon

> **Note**
>
> OpenSearch dan peningkatan versi Elasticsearch berbeda dari pembaruan perangkat lunak layanan. Untuk informasi tentang memperbarui perangkat lunak layanan untuk domain OpenSearch Layanan Anda, lihatthe section called “Pembaruan perangkat lunak layanan”.

Amazon OpenSearch Service menawarkan peningkatan di tempat untuk domain yang menjalankan OpenSearch 1.0 atau yang lebih baru, atau Elasticsearch 5.1 atau versi lebih baru. Jika Anda menggunakan layanan seperti Amazon Data Firehose atau Amazon CloudWatch Logs untuk mengalirkan data ke OpenSearch Layanan, periksa apakah layanan ini mendukung versi yang lebih baru sebelum bermigrasi. OpenSearch

Topik

- Jalur pemutakhiran yang didukung
- Memulai upgrade (konsol)
- Memulai upgrade (CLI)
- Memulai upgrade (SDK)
- Memecahkan masalah kegagalan validasi
- Memecahkan masalah peningkatan
- Menggunakan snapshot untuk memigrasi data

Jalur pemutakhiran yang didukung

Saat ini, OpenSearch Layanan mendukung jalur pemutakhiran berikut:

Dari versi	Ke versi
OpenSearch 1.3 atau 2. x	OpenSearch 2. x Versi 2.3 memiliki perubahan melanggar berikut: • `type`Parameter telah dihapus dari semua titik akhir OpenSearch API di versi 2.0. Untuk informasi selengkapnya, lihat perubahan yang melanggar. • Jika domain Anda berisi indeks apa pun (panas, UltraWarm, atau dingin) yang awalnya dibuat di Elasticsearch 6.8, indeks tersebut tidak kompatibel dengan 2.3. OpenSearch Sebelum Anda meningkatkan ke versi 2.3, Anda harus mengindeks ulang indeks yang tidak kompatibel. Untuk indeks yang tidak kompatibel UltraWarm atau dingin, migrasikan ke penyimpanan panas, indeks ulang data, lalu migrasikan kembali ke penyimpanan hangat atau dingin. Bergantian, Anda dapat menghapus indeks jika Anda tidak lagi membutuhkannya. Jika Anda secara tidak sengaja memutakhirkan domain ke versi 2.3 tanpa melakukan langkah-langkah ini terlebih dahulu, Anda tidak akan dapat memigrasikan indeks yang tidak kompatibel dari tingkat penyimpanannya saat ini. Satu-satunya pilihan Anda adalah menghapusnya.
OpenSearch 1. x	OpenSearch 1. x
Elasticsearch 7. x	Elasticsearch 7. x atau OpenSearch 1. x **Important** OpenSearch 1. x memperkenalkan banyak perubahan yang melanggar. Untuk detailnya, lihat Ganti nama OpenSearch Layanan Amazon.
Elasticsearch 6.8	Elasticsearch 7. x atau OpenSearch 1. x

Dari versi	Ke versi
	Important Elasticsearch 7.0 dan OpenSearch 1.0 mencakup banyak perubahan yang melanggar. Sebelum memulai peningkatan di tempat, kami sarankan untuk mengambil snapshot manual dari 6. x domain, memulihkannya pada tes 7. x atau OpenSearch 1. x domain, dan menggunakan domain uji itu untuk mengidentifikasi potensi masalah peningkatan. Untuk melanggar perubahan di OpenSearch 1.0, lihatGanti nama OpenSearch Layanan Amazon. Seperti Elasticsearch 6. x, indeks hanya dapat berisi satu jenis pemetaan, tetapi tipe itu sekarang harus diberi nama. `_doc` Akibatnya, API tertentu tidak lagi memerlukan jenis pemetaan di isi permintaan (seperti API `_bulk`). Untuk indeks baru, Elasticsearch 7 yang dihosting sendiri. x dan OpenSearch 1. x memiliki jumlah pecahan default satu. OpenSearch Domain layanan di Elasticsearch 7. x dan kemudian mempertahankan default sebelumnya dari lima.
Elasticsearch 6. x	Elasticsearch 6. x
Elasticsearch 5.6	Elasticsearch 6. x **Important** Indeks dibuat dalam versi 6. x tidak lagi mendukung beberapa jenis pemetaan. Indeks dibuat dalam versi 5. x masih mendukung beberapa jenis pemetaan saat dikembalikan ke 6. x gugus. Periksa apakah kode klien Anda hanya membuat satu jenis pemetaan per indeks. Untuk meminimalkan waktu henti selama pemutakhiran dari Elasticsearch 5.6 ke 6. x, OpenSearch Service mengindeks ulang `.kibana` indeks ke`.kibana-6` , menghapus, membuat alias bernama `.kibana.kibana`, dan memetakan indeks baru ke alias baru.

Dari versi	Ke versi
Elasticsearch 5. x	Elasticsearch 5. x

Proses peningkatan terdiri dari tiga langkah:

1. Pemeriksaan pra-peningkatan — OpenSearch Layanan memeriksa masalah yang dapat memblokir peningkatan dan tidak melanjutkan ke langkah berikutnya kecuali pemeriksaan ini berhasil.
2. Snapshot — OpenSearch Layanan mengambil snapshot dari cluster OpenSearch atau Elasticsearch dan tidak melanjutkan ke langkah berikutnya kecuali snapshot berhasil. Jika pemutakhiran gagal, OpenSearch Service menggunakan snapshot ini untuk mengembalikan cluster ke keadaan semula. Untuk informasi selengkapnya, lihat the section called “Tidak dapat menurunkan versi setelah peningkatan”.
3. Upgrade - OpenSearch Layanan memulai upgrade, yang dapat memakan waktu dari 15 menit hingga beberapa jam untuk menyelesaikannya. OpenSearch Dasbor mungkin tidak tersedia selama beberapa atau semua peningkatan.

Memulai upgrade (konsol)

Proses pemutakhiran tidak dapat diubah dan tidak dapat dijeda atau dibatalkan. Selama peningkatan, Anda tidak dapat membuat perubahan konfigurasi pada domain. Sebelum memulai peningkatan, periksa kembali apakah Anda ingin melanjutkan. Anda dapat menggunakan langkah-langkah yang sama ini untuk melakukan pemeriksaan pra-peningkatan tanpa benar-benar memulai peningkatan.

Jika cluster memiliki node master khusus, OpenSearch upgrade selesai tanpa downtime. Jika tidak, klaster mungkin tidak responsif untuk beberapa detik pasca-peningkatan saat klaster memilih simpul utama.

Untuk meng-upgrade domain ke versi yang lebih baru dari OpenSearch atau Elasticsearch

1. Ambil snapshot manual dari domain Anda. Snapshot ini berfungsi sebagai cadangan yang dapat Anda pulihkan pada domain baru jika Anda ingin kembali menggunakan OpenSearch versi sebelumnya.
2. Masuk ke https://aws.amazon.com, dan pilih Masuk ke Konsol.

3. Di bawah Analytics, pilih OpenSearch Layanan Amazon.
4. Di panel navigasi, di bawah Domain, pilih domain yang ingin Anda tingkatkan.
5. Pilih Tindakan dan Tingkatkan.
6. Pilih versi untuk meng-upgrade ke. Jika Anda memutakhirkan ke OpenSearch versi, opsi Aktifkan mode kompatibilitas akan muncul. Jika Anda mengaktifkan pengaturan ini, OpenSearch laporkan versinya sebagai 7.10 untuk memungkinkan klien dan plugin Elasticsearch OSS seperti Logstash untuk terus bekerja dengan Amazon Service. OpenSearch Anda dapat menonaktifkan pengaturan ini nanti
7. Pilih Tingkatkan.
8. Periksa Status di dasbor domain untuk memantau status peningkatan.

Memulai upgrade (CLI)

Anda dapat menggunakan operasi berikut untuk mengidentifikasi versi OpenSearch atau Elasticsearch yang benar untuk domain Anda, memulai pemutakhiran di tempat, melakukan pemeriksaan pra-pemutakhiran, dan melihat kemajuan:

- `get-compatible-versions` (`GetCompatibleVersions`)
- `upgrade-domain` (`UpgradeDomain`)
- `get-upgrade-status` (`GetUpgradeStatus`)
- `get-upgrade-history` (`GetUpgradeHistory`)

Untuk informasi selengkapnya, lihat referensi perintahAWS CLI dan Referensi API OpenSearch Layanan Amazon.

Memulai upgrade (SDK)

Contoh ini menggunakan klien Python OpenSearchServicetingkat rendah dari AWS SDK for Python (Boto) untuk memeriksa apakah domain memenuhi syarat untuk ditingkatkan ke versi tertentu, memutakhirkannya, dan terus memeriksa status pemutakhiran.

```
import boto3
from botocore.config import Config
import time

# Build the client using the default credential configuration.
```

```
# You can use the CLI and run 'aws configure' to set access key, secret
# key, and default Region.

DOMAIN_NAME = ''  # The name of the domain to upgrade
TARGET_VERSION = ''  # The version you want to upgrade the domain to. For example,
 OpenSearch_1.1

my_config = Config(
    # Optionally lets you specify a Region other than your default.
    region_name='us-east-1'
)
client = boto3.client('opensearch', config=my_config)


def check_versions():
    """Determine whether domain is eligible for upgrade"""
    response = client.get_compatible_versions(
        DomainName=DOMAIN_NAME
    )
    compatible_versions = response['CompatibleVersions']
    for i in range(len(compatible_versions)):
        if TARGET_VERSION in compatible_versions[i]["TargetVersions"]:
            print('Domain is eligible for upgrade to ' + TARGET_VERSION)
            upgrade_domain()
            print(response)
        else:
            print('Domain not eligible for upgrade to ' + TARGET_VERSION)


def upgrade_domain():
    """Upgrades the domain"""
    response = client.upgrade_domain(
        DomainName=DOMAIN_NAME,
        TargetVersion=TARGET_VERSION
    )
    print('Upgrading domain to ' + TARGET_VERSION + '...' + response)
    time.sleep(5)
    wait_for_upgrade()


def wait_for_upgrade():
    """Get the status of the upgrade"""
    response = client.get_upgrade_status(
        DomainName=DOMAIN_NAME
```

```
    )
    if (response['UpgradeStep']) == 'UPGRADE' and (response['StepStatus']) ==
 'SUCCEEDED':
        print('Domain successfully upgraded to ' + TARGET_VERSION)
    elif (response['StepStatus']) == 'FAILED':
        print('Upgrade failed. Please try again.')
    elif (response['StepStatus']) == 'SUCCEEDED_WITH_ISSUES':
        print('Upgrade succeeded with issues')
    elif (response['StepStatus']) == 'IN_PROGRESS':
        time.sleep(30)
        wait_for_upgrade()


def main():
    check_versions()


if __name__ == "__main__":
    main()
```

Memecahkan masalah kegagalan validasi

Saat Anda memulai pemutakhiran versi OpenSearch atau Elasticsearch, OpenSearch Layanan terlebih dahulu melakukan serangkaian pemeriksaan validasi untuk memastikan bahwa domain Anda memenuhi syarat untuk peningkatan. Jika salah satu pemeriksaan ini gagal, Anda menerima pemberitahuan yang berisi masalah spesifik yang harus Anda perbaiki sebelum memutakhirkan domain Anda. Untuk daftar potensi masalah dan langkah-langkah untuk menyelesaikannya, lihatthe section called “Memecahkan masalah kesalahan validasi”.

Memecahkan masalah peningkatan

Upgrade di tempat membutuhkan domain yang sehat. Domain Anda mungkin tidak memenuhi syarat untuk peningkatan atau gagal ditingkatkan karena berbagai alasan. Tabel berikut menunjukkan masalah yang paling umum.

Masalah	Deskripsi
Plugin opsional tidak didukung	Ketika Anda meng-upgrade domain dengan plugin opsional, OpenSearch Layanan secara otomatis meningkatkan plugin juga. Oleh karena itu, versi target untuk domain Anda juga harus mendukung plugin opsional

Masalah	Deskripsi
	ini. Jika domain memiliki plugin opsional diinstal yang tidak tersedia untuk versi target, permintaan upgrade gagal.
Terlalu banyak serpihan per simpul	OpenSearch, serta 7. x versi Elasticsearch, memiliki pengaturan default tidak lebih dari 1.000 pecahan per node. Jika node di kluster Anda saat ini melebihi pengaturan ini, OpenSearch Service tidak mengizinkan Anda untuk meningkatkan. Lihat the section called “Melampaui batas serpihan maksimum” untuk opsi pemecahan masalah.
Domain sedang diproses	Domain sedang di tengah-tengah perubahan konfigurasi. Periksa kelayakan peningkatan setelah operasi selesai.
Status klaster merah	Satu atau lebih indeks di cluster berwarna merah. Untuk langkah-langkah pemecahan masalah, lihat the section called “Status klaster merah”.
Tingkat kesalahan tinggi	Cluster mengembalikan sejumlah besar kesalahan 5 xx saat mencoba memproses permintaan. Masalah ini biasanya merupakan hasil dari terlalu banyaknya permintaan baca atau tulis secara bersamaan. Pertimbangkan untuk mengurangi lalu lintas ke klaster atau menskalakan domain Anda.
Split brain	Otak terbelah berarti bahwa cluster Anda memiliki lebih dari satu simpul master dan telah dibagi menjadi dua cluster yang tidak akan pernah bergabung kembali dengan sendirinya. Anda dapat menghindari split brain dengan menggunakan jumlah yang direkomendasikan untuk simpul utama terdedikasi. Untuk bantuan pemulihan dari split brain, hubungi AWS Support.
Simpul utama tidak ditemukan	OpenSearch Layanan tidak dapat menemukan simpul master cluster. Jika domain Anda menggunakan multi-AZ, kegagalan Availability Zone mungkin telah menyebabkan klaster kehilangan kuorum dan tidak dapat memilih simpul utama yang baru. Jika masalah tidak teratasi sendiri, hubungi AWS Support.
Terlalu banyak tugas yang tertunda	Simpul utama berada di bawah beban berat dan memiliki banyak tugas yang tertunda. Pertimbangkan untuk mengurangi lalu lintas ke klaster atau menskalakan domain Anda.

Masalah	Deskripsi
Volume penyimpanan yang terganggu	Volume disk dari satu atau lebih simpul tidak berfungsi dengan benar. Masalah ini sering terjadi bersamaan dengan masalah lainnya, seperti tingkat kesalahan yang tinggi atau terlalu banyak tugas yang tertunda. Jika terjadi dalam isolasi dan tidak teratasi sendiri, hubungi AWS Support.
Masalah kunci KMS	Kunci KMS yang digunakan untuk mengenkripsi domain tidak dapat diakses atau hilang. Untuk informasi selengkapnya, lihat the section called “Memantau domain yang mengenkripsi data saat tidak digunakan”.
Snapshot sedang berlangsung	Domain saat ini sedang mengambil snapshot. Periksa kelayakan peningkatan setelah snapshot selesai. Juga periksa apakah Anda dapat mencantumkan repositori snapshot manual, mencantumkan snapshot dalam repositori tersebut, dan mengambil snapshot manual. Jika OpenSearch Layanan tidak dapat memeriksa apakah snapshot sedang berlangsung, pemutakhiran dapat gagal.
Waktu habis atau kegagalan snapshot	Snapshot pra-peningkatan terlalu lama untuk diselesaikan atau gagal. Periksa kesehatan klaster, dan coba lagi. Jika masalah berlanjut, hubungi AWS Support.
Indeks yang tidak kompatibel	Satu atau lebih indeks tidak kompatibel dengan versi target. Masalah ini dapat terjadi jika Anda memigrasikan indeks dari versi lama OpenSearch atau Elasticsearch. Indeks ulang indeks dan coba lagi.
Penggunaan disk yang tinggi	Penggunaan disk untuk klaster di atas 90%. Hapus data atau skalakan domain, dan coba lagi.
Penggunaan JVM yang tinggi	Tekanan memori JVM di atas 75%. Kurangi lalu lintas ke klaster atau skalakan domain, dan coba lagi.
OpenSearch Masalah alias dasbor	`.dashboards` sudah dikonfigurasi sebagai alias dan memetakan ke indeks yang tidak kompatibel, kemungkinan dari versi Dasbor sebelumny a. OpenSearch Index ulang dan coba lagi.

Masalah	Deskripsi
Status Dasbor Merah	OpenSearch Status dasbor berwarna merah. Coba gunakan Dasbor saat pemutakhiran selesai. Jika status merah tetap ada, selesaikan secara manual, dan coba lagi.
Kompatibilitas lintas klaster	Anda hanya dapat memutakhirkan jika kompatibilitas lintas cluster dipertahankan antara domain sumber dan tujuan setelah pemutakhiran. Selama proses peningkatan, koneksi apa pun yang tidak kompatibel akan diidentifikasi. Untuk melanjutkan, tingkatkan domain jarak jauh atau hapus koneksi yang tidak kompatibel. Perhatikan bahwa jika replikasi aktif di domain, Anda tidak dapat melanjutkannya setelah Anda menghapus koneksi.
Masalah OpenSearch layanan layanan lainnya	Masalah dengan OpenSearch Layanan itu sendiri dapat menyebabkan domain Anda ditampilkan sebagai tidak memenuhi syarat untuk peningkatan. Jika tidak ada syarat sebelumnya yang berlaku untuk domain Anda dan masalah berlanjut selama lebih dari satu hari, hubungi AWS Support.

Menggunakan snapshot untuk memigrasi data

Peningkatan di tempat adalah cara yang lebih mudah, lebih cepat, dan lebih andal untuk meningkatkan domain ke versi yang lebih baru OpenSearch atau Elasticsearch. Snapshot adalah opsi yang baik jika Anda perlu bermigrasi dari versi Elasticsearch sebelum 5.1 atau ingin bermigrasi ke klaster yang sepenuhnya baru.

Tabel berikut menunjukkan cara menggunakan snapshot untuk memigrasikan data ke domain yang menggunakan versi berbeda OpenSearch atau Elasticsearch. Untuk informasi selengkapnya tentang mengambil dan memulihkan snapshots, lihat the section called “Membuat snapshot indeks”.

Dari versi	Ke versi	Proses migrasi
OpenSearch 1.3 atau 2. x	OpenSearch 2. x	1. Tinjau perubahan yang melanggar untuk OpenSearch 2.3 untuk melihat apakah Anda perlu melakukan penyesuaian pada indeks atau aplikasi Anda. 2. Buat snapshot manual dari 1.3 atau 2. x domain.

Dari versi	Ke versi	Proses migrasi

		3. Buat 2. x domain yang merupakan versi yang lebih tinggi dari 1.3 atau 2 asli Anda. x domain.
		4. Kembalikan snapshot dari domain asli ke 2. x domain. Selama operasi, Anda mungkin perlu mengembalikan `.opensearch` indeks Anda dengan nama baru:
		`POST _snapshot/ <repository-name> /<snapshot-name>/_restore`
		`{`
		`"indices": "*",`
		`"ignore_unavailable": true,`
		`"rename_pattern": ".opensearch",`
		`"rename_replacement": ".backup-opensearch"`
		`}`
		Kemudian Anda dapat mengindeks ulang `.backup-opensearch` pada domain dan alias baru ke `.opensearch` . Perhatikan bahwa panggilan `_restore` REST tidak termasuk `include_global_state` karena default in `_restore` adalah false. Akibatnya, domain pengujian tidak akan menyertakan templat indeks apa pun dan tidak akan memiliki status lengkap dari cadangan.
		5. Jika Anda tidak lagi memerlukan domain asli Anda, hapus domain tersebut. Jika tidak, Anda akan terus dikenakan biaya untuk domain tersebut.

Dari versi	Ke versi	Proses migrasi

OpenSearch 1. x	OpenSearch 1. x	1. Buat snapshot manual dari 1. x domain.
		2. Buat 1. x domain yang merupakan versi yang lebih tinggi dari domain asli Anda 1. x domain.
		3. Kembalikan snapshot dari domain asli ke yang baru 1. x domain. Selama operasi, Anda mungkin perlu mengembalikan `.opensearch` indeks Anda dengan nama baru:
		`POST _snapshot/ <repository-name> /<snapshot-name>/_restore`
		`{`
		`  "indices": "*",`
		`  "ignore_unavailable": true,`
		`  "rename_pattern": ".opensearch",`
		`  "rename_replacement": ".backup-opensearch"`
		`}`
		Kemudian Anda dapat mengindeks ulang `.backup-opensearch` pada domain dan alias baru ke `.opensearch` . Perhatikan bahwa panggilan `_restore` REST tidak termasuk `include_global_state` karena default in `_restore` adalah false. Akibatnya, domain pengujian tidak akan menyertakan templat indeks apa pun dan tidak akan memiliki status lengkap dari cadangan.
		4. Jika Anda tidak lagi memerlukan domain asli Anda, hapus domain tersebut. Jika tidak, Anda akan terus dikenakan biaya untuk domain tersebut.

Dari versi	Ke versi	Proses migrasi

Elasticsearch 6. x atau 7. x	OpenSearch 1. x	1. Tinjau perubahan yang melanggar untuk OpenSearc h 1.0 untuk melihat apakah Anda perlu melakukan penyesuaian pada indeks atau aplikasi Anda.
		2. Buat snapshot manual dari Elasticsearch 7. x atau 6. x domain.
		3. Buat OpenSearch 1. x domain.
		4. Kembalikan snapshot dari domain Elasticsearch ke domain. OpenSearch Selama operasi, Anda mungkin perlu mengembalikan `.elasticsearch` indeks Anda dengan nama baru:
		`POST _snapshot/ <repository-name> /<snapshot-name>/_restore`
		`{`
		`  "indices": "*",`
		`  "ignore_unavailable": true,`
		`  "rename_pattern": ".elasticsearch",`
		`  "rename_replacement": ".backup-opensearc h"`
		`}`
		Kemudian Anda dapat mengindeks ulang `.backup-opensearch` pada domain dan alias baru ke `.elasticsearch` . Perhatikan bahwa panggilan `_restore` REST tidak termasuk `include_g lobal_state` karena default in `_restore` adalah false. Akibatnya, domain pengujian tidak akan menyertakan templat indeks apa pun dan tidak akan memiliki status lengkap dari cadangan.
		5. Jika Anda tidak lagi memerlukan domain asli Anda, hapus domain tersebut. Jika tidak, Anda akan terus dikenakan biaya untuk domain tersebut.

Dari versi	Ke versi	Proses migrasi

Elasticsearch 6.x	Elasticsearch 7.x	1. Tinjau perubahan yang melanggar untuk 7.0 untuk melihat apakah Anda perlu melakukan penyesuaian pada indeks atau aplikasi Anda.
		2. Buat snapshot manual dari domain 6.x.
		3. Buat domain 7.x.
		4. Memulihkan snapshot dari domain asli ke domain 7.x. Selama operasi, Anda mungkin perlu untuk memulihkan indeks `.opensearch` dengan nama baru:
		`POST _snapshot/ <repository-name> /<snapshot-name>/_restore`
		`{`
		`  "indices": "*",`
		`  "ignore_unavailable": true,`
		`  "rename_pattern": ".elasticsearch",`
		`  "rename_replacement": ".backup-elasticsearch"`
		`}`
		Kemudian Anda dapat mengindeks ulang `.backup-elasticsearch` pada domain dan alias baru ke `.elasticsearch` . Perhatikan bahwa panggilan `_restore` REST tidak termasuk `include_global_state` karena default in `_restore` adalah false. Akibatnya, domain pengujian tidak akan menyertakan templat indeks apa pun dan tidak akan memiliki status lengkap dari cadangan.
		5. Jika Anda tidak lagi memerlukan domain asli Anda, hapus domain tersebut. Jika tidak, Anda akan terus dikenakan biaya untuk domain tersebut.

Dari versi	Ke versi	Proses migrasi
Elasticsearch 6.x	Elasticsearch 6.8	1. Buat snapshot manual dari domain 6.x. 2. Buat domain 6.8. 3. Pulihkan snapshot dari domain asli ke domain 6.8. 4. Jika Anda tidak lagi memerlukan domain asli Anda, hapus domain tersebut. Jika tidak, Anda akan terus dikenakan biaya untuk domain tersebut.
Elasticsearch 5.x	Elasticsearch 6.x	1. Tinjau perubahan besar pada 6.0 untuk melihat apakah Anda perlu melakukan penyesuaian pada indeks atau aplikasi Anda. 2. Buat snapshot manual dari domain 5.x. 3. Buat domain 6.x. 4. Pulihkan snapshot dari domain asli ke domain 6.x. 5. Jika Anda tidak lagi memerlukan domain 5.x Anda, hapus domain tersebut. Jika tidak, Anda akan terus dikenakan biaya untuk domain tersebut.
Elasticsearch 5.x	Elasticsearch 5.6	1. Buat snapshot manual dari domain 5.x. 2. Buat domain 5.6. 3. Pulihkan snapshot dari domain asli ke domain 5.6. 4. Jika Anda tidak lagi memerlukan domain asli Anda, hapus domain tersebut. Jika tidak, Anda akan terus dikenakan biaya untuk domain tersebut.
Elasticsearch 2.3	Elasticsearch 6.x	Snapshot Elasticsearch 2.3 tidak kompatibel dengan 6.x. Untuk memigrasikan data Anda secara langsung dari 2.3 ke 6. x, Anda harus secara manual membuat ulang indeks Anda di domain baru. Bergantian, Anda dapat mengikuti 2.3 hingga 5. x langkah dalam tabel ini, melakukan `_reindex` operasi di 5 baru. x domain untuk mengonversi indeks 2.3 Anda menjadi 5. x indeks, dan kemudian ikuti 5. x sampai 6. x langkah.

Dari versi	Ke versi	Proses migrasi
Elasticsearch 2.3	Elasticsearch 5.x	1. Tinjau perubahan yang melanggar untuk 5.0 untuk melihat apakah Anda perlu melakukan penyesuaian pada indeks atau aplikasi Anda. 2. Buat snapshot manual dari domain 2.3. 3. Buat domain 5.x. 4. Pulihkan snapshot dari domain 2.3 ke domain 5.x. 5. Jika Anda tidak lagi memerlukan domain 2.3 Anda, hapus domain tersebut. Jika tidak, Anda akan terus dikenakan biaya untuk domain tersebut.
Elasticsearch 1.5	Elasticsearch 5.x	Snapshot Elasticsearch 1.5 tidak kompatibel dengan 5.x. Untuk memigrasikan data Anda dari 1,5 ke 5. x, Anda harus secara manual membuat ulang indeks Anda di domain baru. **Important** 1.5 snapshot kompatibel dengan 2.3, tetapi domain OpenSearch Layanan 2.3 tidak mendukung operasi. `_reindex` Karena Anda tidak dapat mengindeks ulang mereka, indeks yang berasal dari domain 1.5 masih gagal memulihkan dari 2.3 snapshot ke 5. x domain.

Dari versi	Ke versi	Proses migrasi
Elasticsearch 1.5	Elasticsearch 2.3	1. Gunakan plugin migrasi untuk mengetahui apakah Anda dapat langsung meningkatkan ke versi 2.3. Anda mungkin perlu melakukan perubahan pada data Anda sebelum migrasi. a. Di peramban web, buka `http://domain-endpoint /_plugin/migration/` . b. Pilih Jalankan pemeriksaan sekarang. c. Tinjau hasilnya dan, jika diperlukan, ikuti petunjuk untuk membuat perubahan pada data Anda. 2. Buat snapshot manual dari domain 1.5. 3. Buat domain 2.3. 4. Pulihkan snapshot dari domain 1.5 ke domain 2.3. 5. Jika Anda tidak lagi memerlukan domain 1.5 Anda, hapus domain tersebut. Jika tidak, Anda akan terus dikenakan biaya untuk domain tersebut.

Membuat titik akhir khusus untuk Amazon Service OpenSearch

Membuat endpoint khusus untuk domain Amazon OpenSearch Service memudahkan Anda untuk merujuk ke URL OpenSearch dan OpenSearch Dasbor Anda. Anda dapat memasukkan branding perusahaan Anda atau hanya menggunakan easier-to-remember titik akhir yang lebih pendek dari yang standar.

Jika Anda perlu beralih ke domain baru, cukup perbarui DNS Anda untuk mengarahkan ke URL baru dan lanjutkan menggunakan titik akhir yang sama seperti sebelumnya.

Anda mengamankan titik akhir kustom dengan membuat sertifikat di AWS Certificate Manager (ACM) atau mengimpor salah satu sertifikat Anda sendiri.

Titik akhir khusus untuk domain baru

Anda dapat mengaktifkan titik akhir kustom untuk domain OpenSearch Layanan baru menggunakan konsol OpenSearch LayananAWS CLI, atau API konfigurasi.

Untuk menyesuaikan titik akhir Anda (konsol)

1. Dari konsol OpenSearch Layanan, pilih Buat domain dan berikan nama untuk domain tersebut.
2. Pada Titik akhir kustom, pilih Aktifkan titik akhir kustom.
3. Untuk Hostname kustom, masukkan nama host titik akhir kustom pilihan Anda. Nama host harus merupakan nama domain yang memenuhi syarat (FQDN), seperti www.yourdomain.com atau example.yourdomain.com.

   > **Note**
   >
   > Jika Anda tidak memilikinya sertifikat wildcard Anda harus mendapatkan sertifikat baru untuk subdomain titik akhir kustom Anda.

4. Untuk AWS sertifikat, pilih sertifikat SSL yang akan digunakan untuk domain Anda. Jika sertifikat tidak tersedia, Anda dapat mengimpor salah satu ke ACM atau menggunakan ACM untuk menyediakan satu sertifikat. Untuk informasi lebih lanjut, lihat Menerbitkan dan Mengelola Sertifikat dalam AWSPanduan Pengguna Certificate Manager.

   > **Note**
   >
   > Sertifikat harus memiliki nama titik akhir kustom dan berada di akun yang sama dengan domain OpenSearch Layanan Anda. Status sertifikat harus DIKELUARKAN.

   - Ikuti langkah-langkah lainnya untuk membuat domain Anda dan pilih Buat.
   - Pilih domain setelah selesai diproses untuk melihat titik akhir kustom Anda.

   Untuk menggunakan CLI atau API konfigurasi, gunakan operasi `CreateDomain` dan `UpdateDomainConfig`. Untuk informasi selengkapnya, lihat Referensi AWS CLIPerintah dan Referensi API Amazon OpenSearch Service.

Titik akhir khusus untuk domain yang sudah ada

Untuk menambahkan titik akhir kustom ke domain OpenSearch Layanan yang ada, pilih Edit dan lakukan langkah 2-4 di atas.

Langkah selanjutnya

Setelah mengaktifkan titik akhir khusus untuk domain OpenSearch Layanan, Anda harus membuat pemetaan CNAME di Amazon Route 53 (atau penyedia layanan DNS pilihan Anda). Anda melakukan ini untuk merutekan lalu lintas ke titik akhir kustom dan subdomainnya. Tanpa pemetaan ini, titik akhir kustom Anda tidak akan bekerja. Untuk langkah-langkah untuk membuat pemetaan ini di Route 53, lihat Mengonfigurasi perutean DNS untuk domain baru dan Membuat zona yang di-hosting untuk subdomain. Untuk penyedia lain, konsultasikan dokumentasi mereka.

Buat catatan CNAME yang mengarahkan titik akhir kustom ke titik akhir domain yang dibuat secara otomatis. Jika domain Anda adalah tumpukan ganda, Anda dapat mengarahkan catatan CNAME Anda ke salah satu dari dua titik akhir yang dihasilkan layanan. Kemampuan tumpukan ganda dari titik akhir kustom bergantung pada titik akhir yang dihasilkan layanan yang Anda arahkan ke catatan CNAME. Nama host endpoint kustom adalah nama catatan CNAME, dan nama host titik akhir domain adalah nilai catatan CNAME.

Jika Anda menggunakan otentikasi SAMP untuk OpenSearch Dasbor, Anda harus memperbarui IDP Anda dengan URL SSO baru.

Auto-Tune untuk Layanan Amazon OpenSearch

Auto-Tune in Amazon OpenSearch Service menggunakan metrik kinerja dan penggunaan dari OpenSearch klaster Anda untuk menyarankan perubahan konfigurasi terkait memori, termasuk ukuran antrian dan cache serta pengaturan Java virtual machine (JVM) pada node Anda. Perubahan opsional ini meningkatkan kecepatan dan stabilitas klaster.

Beberapa perubahan segera diterapkan, sementara yang lain dijadwalkan selama jendela off-peak domain Anda. Anda dapat kembali ke pengaturan OpenSearch Layanan default kapan saja. Saat Auto-Tune mengumpulkan dan menganalisis metrik kinerja untuk domain Anda, Anda dapat melihat rekomendasinya di konsol OpenSearch Layanan di halaman Pemberitahuan.

Auto-Tune tersedia secara komersial Wilayah AWS di domain yang menjalankan OpenSearch versi apa pun, atau Elasticsearch 6.7 atau yang lebih baru, dengan jenis instans yang didukung.

Topik

- Jenis perubahan
- Mengaktifkan atau menonaktifkan Auto-Tune

- Penjadwalan penyempurnaan Auto-Tune
- Memantau perubahan Auto-Tune

Jenis perubahan

Auto-Tune memiliki dua kategori besar perubahan:

- Perubahan nondisruptive yang diterapkan saat cluster berjalan.
- Perubahan yang memerlukan penerapan biru/hijau, yang berlaku selama jendela off-peak domain.

Berdasarkan metrik kinerja domain Anda, Auto Tune dapat menyarankan penyesuaian pada pengaturan berikut:

Jenis perubahan	Kategori	Deskripsi
Ukuran tumpukan JVM	Biru/hijau	Secara default, OpenSearch Service menggunakan 50% RAM instans untuk heap JVM, hingga ukuran heap 32 GiB. Meningkatkan persentase ini memberi OpenSearch lebih banyak memori, tetapi menyisakan lebih sedikit untuk sistem operasi dan proses lainnya. Nilai yang lebih besar dapat mengurangi jumlah jeda pengumpulan sampah, tetapi menambah panjang jeda tersebut.
Pengaturan generasi muda JVM	Biru/hijau	Pengaturan “generasi muda” JVM menjejaskan kekerapan koleksi sampah minor. Koleksi minor yang lebih sering dapat mengurangi jumlah koleksi utama dan jeda.
Ukuran antrian	Tidak mengganggu	Secara default, ukuran antrian pencarian adalah `1000` dan ukuran antrian tulis adalah `10000`. Auto-Tune secara otomatis menskalakan pencarian dan menulis antrian jika tumpukan tambahan tersedia untuk menangani permintaan.
Ukuran cache	Tidak mengganggu	Cache bidang memonitor struktur data tumpukan, jadi penting untuk memantau penggunaan cache. Auto-Tune menskalakan

Jenis perubahan	Kategori	Deskripsi
		ukuran cache data lapangan untuk menghindari masalah kehabisan memori dan pemutus sirkuit. Cache permintaan serpihan dikelola pada tingkat simpul dan memiliki ukuran maksimum default 1% dari tumpukan. Auto-Tune menskalakan ukuran cache permintaan serpihan untuk menerima lebih banyak permintaan pencarian dan indeks daripada yang dapat ditangani oleh klaster yang dikonfigurasi.
Ukuran permintaan	Tidak mengganggu	Secara default, ketika ukuran agregat permintaan dalam penerbangan melampaui 10% dari total JVM (2% untuk tipe `t2` instans dan 1% untuk`t3.small`), OpenSearch membatasi semua permintaan baru `_search` dan `_bulk` permintaan hingga permintaan yang ada selesai. Auto-Tune secara otomatis menyetel ambang batas ini, biasanya antara 5-15%, berdasarkan jumlah JVM yang saat ini ditempati pada sistem. Misalnya, jika tekanan memori JVM tinggi, Auto-Tune dapat mengurangi ambang batas menjadi 5%, di mana Anda mungkin melihat lebih banyak penolakan hingga cluster stabil dan ambang batas meningkat.

Mengaktifkan atau menonaktifkan Auto-Tune

OpenSearch Layanan mengaktifkan Auto-Tune secara default pada domain baru. Untuk mengaktifkan atau menonaktifkan Auto-Tune pada domain yang ada, kami sarankan menggunakan konsol, yang menyederhanakan proses. Mengaktifkan Auto-Tune tidak menyebabkan deployment biru/hijau.

Saat ini Anda tidak dapat mengaktifkan atau menonaktifkan Auto-Tune menggunakanAWS CloudFormation.

Konsol

Untuk mengaktifkan Auto-Tune pada domain yang ada

1. Buka konsol OpenSearch Layanan Amazon di https://console.aws.amazon.com/aos/home.
2. Di panel navigasi, di bawah Domain, pilih nama domain untuk membuka konfigurasi cluster.
3. Pilih Aktifkan jika Penyetelan Otomatis belum diaktifkan.
4. Secara opsional, pilih Jendela Off-peak untuk menjadwalkan pengoptimalan yang memerlukan penerapan biru/hijau selama jendela off-peak yang dikonfigurasi domain. Untuk informasi selengkapnya, lihat the section called “Penjadwalan penyempurnaan Auto-Tune”.
5. Pilih Save changes (Simpan perubahan).

CLI

Untuk mengaktifkan Auto-Tune menggunakanAWS CLI, kirim UpdateDomainConfigpermintaan:

```
aws opensearch update-domain-config \
  --domain-name my-domain \
  --auto-tune-options DesiredState=ENABLED
```

Penjadwalan penyempurnaan Auto-Tune

Sebelum 16 Februari 2023, Auto-Tune menggunakan jendela pemeliharaan untuk menjadwalkan perubahan yang memerlukan penerapan biru/hijau. Jendela pemeliharaan sekarang tidak digunakan lagi demi jendela off-peak, yang merupakan blok waktu 10 jam harian di mana domain Anda biasanya mengalami lalu lintas rendah. Anda dapat memodifikasi waktu mulai default untuk jendela off-peak, tetapi Anda tidak dapat mengubah panjangnya.

Setiap domain yang mengaktifkan jendela pemeliharaan Auto-Tune sebelum pengenalan jendela off-peak pada 16 Februari 2023 dapat terus menggunakan jendela pemeliharaan lama tanpa gangguan. Namun, kami menyarankan Anda memigrasikan domain yang ada untuk menggunakan jendela off-peak untuk pemeliharaan domain. Untuk petunjuk, lihat the section called “Migrasi dari jendela pemeliharaan Auto-Tune”.

Konsol

Untuk menjadwalkan tindakan Auto-Tune jendela off-peak

1. Buka konsol OpenSearch Layanan Amazon di https://console.aws.amazon.com/aos/home.

2. Di panel navigasi, di bawah Domain, pilih nama domain untuk membuka konfigurasi cluster.
3. Buka tab Auto-Tune dan pilih Edit.
4. Pilih Aktifkan jika Penyetelan Otomatis belum diaktifkan.
5. Di bawah Jadwalkan pengoptimalan selama jendela off-peak, pilih Jendela Off-peak.
6. Pilih Save changes (Simpan perubahan).

CLI

Untuk mengonfigurasi domain Anda untuk menjadwalkan tindakan Auto-Tune selama jendela off-peak yang dikonfigurasi, sertakan `UseOffPeakWindow` dalam permintaan: UpdateDomainConfig

```
aws opensearch update-domain-config \
  --domain-name my-domain \
  --auto-tune-options
 DesiredState=ENABLED,UseOffPeakWindow=true,MaintenanceSchedules=null
```

Memantau perubahan Auto-Tune

Anda dapat memantau statistik Auto-Tune diAmazon CloudWatch. Untuk daftar lengkap metrik, lihat the section called “Metrik Penyetelan Otomatis”.

OpenSearch Layanan mengirimkan acara Auto-Tune ke Amazon EventBridge. Anda dapat menggunakan EventBridge untuk mengonfigurasi aturan yang mengirim email atau melakukan tindakan tertentu saat acara diterima. Untuk melihat format setiap acara Auto-Tune yang dikirimkan EventBridge, lihatthe section called “Peristiwa Auto-Tune”.

Menandai domain OpenSearch Layanan Amazon

Tag memungkinkan Anda menetapkan informasi arbitrer ke domain OpenSearch Layanan Amazon sehingga Anda dapat mengkategorikan dan memfilter informasi tersebut. Tag adalah pasangan kunci-nilai yang Anda tentukan dan kaitkan dengan domain OpenSearch Layanan. Anda dapat menggunakan tag ini untuk melacak biaya dengan mengelompokkan biaya untuk sumber daya yang ditandai serupa. AWS tidak menerapkan makna semantik apa pun pada tag Anda. Tanda ditafsirkan dengan ketat sebagai string karakter. Semua tanda memiliki elemen berikut:

Elemen tanda	Deskripsi	Diperlukan
Kunci tanda	Kunci tanda adalah nama dari tanda. Kunci harus unik untuk domain OpenSearch Layanan yang dilampirkan. Untuk daftar batasan dasar pada kunci dan nilai tanda, lihat Pembatasan Tanda yang Ditetapkan Pengguna.	Ya
Nilai tanda	Nilai tanda adalah nilai string dari tanda. Nilai tanda dapat berupa `null` dan tidak harus unik dalam kumpulan tanda. Misalnya, Anda dapat memiliki pasangan nilai kunci dalam kumpulan tanda project/Trinity dan cost-center/Trinity. Untuk daftar batasan dasar pada kunci dan nilai tanda, lihat Pembatasan Tanda yang Ditetapkan Pengguna.	Tidak

Setiap domain OpenSearch Layanan memiliki kumpulan tag, yang berisi semua tag yang ditetapkan ke domain OpenSearch Layanan tersebut. AWS tidak secara otomatis menetapkan tag apa pun ke domain OpenSearch Layanan. Sebuah kumpulan tanda dapat berisi antara 0 dan 50 tanda. Jika Anda menambahkan tanda ke domain dengan kunci yang sama dengan tanda yang ada, nilai yang baru akan menimpa nilai yang lama.

Contoh penandaan

Anda dapat menggunakan kunci, untuk menentukan kategori, dan nilai dapat berupa item dalam kategori tersebut. Misalnya, Anda dapat menentukan kunci tag `project` dan nilai tag`Salix`, yang menunjukkan bahwa domain OpenSearch Layanan ditetapkan ke proyek Salix. Anda juga dapat menggunakan tag untuk menunjuk domain OpenSearch Layanan sebagai digunakan untuk pengujian atau produksi dengan menggunakan kunci seperti `environment=test` atau. `environment=production` Coba gunakan sekumpulan kunci tag yang konsisten untuk memudahkan melacak metadata yang terkait dengan domain OpenSearch Layanan.

Anda juga dapat menggunakan tag untuk mengatur AWS tagihan Anda untuk mencerminkan struktur biaya Anda sendiri. Untuk melakukan ini, daftar untuk mendapatkan Akun AWS tagihan Anda dengan nilai kunci tag disertakan. Lalu, kelola informasi penagihan Anda sesuai dengan sumber daya dengan nilai kunci tanda yang sama untuk melihat biaya sumber daya gabungan. Misalnya, Anda dapat menandai beberapa domain OpenSearch Layanan dengan pasangan nilai kunci, lalu mengatur informasi penagihan Anda untuk melihat total biaya untuk setiap domain di beberapa layanan. Untuk

informasi selengkapnya, lihat Penggunaan Tanda Alokasi Biaya dalam dokumentasi Manajemen Penagihan dan BiayaAWS .

> **Note**
>
> Tanda disimpan di cache untuk tujuan otorisasi. Karena itu, penambahan dan pembaruan tag pada domain OpenSearch Layanan mungkin memerlukan waktu beberapa menit sebelum tersedia.

Cara menggunakan tanda (konsol)

Konsol adalah cara termudah untuk menandai domain.

Untuk membuat tag (konsol)

1. Masuk ke https://aws.amazon.com, kemudian pilih Masuk ke Konsol.
2. Di bawah Analytics, pilih OpenSearch Layanan Amazon.
3. Pilih domain yang ingin Anda tambahkan tag dan buka tab Tag.
4. Pilih Kelola dan Tambahkan tag baru.
5. Masukkan kunci tag dan nilai opsional.
6. Pilih Simpan.

Untuk menghapus tag, ikuti langkah yang sama dan pilih Hapus pada halaman Kelola tag.

Untuk informasi selengkapnya tentang menggunakan konsol untuk bekerja dengan tag, lihat Editor Tag di Panduan Memulai KonsolAWS Manajemen.

Cara menggunakan tanda (AWS CLI)

Anda dapat membuat tag sumber daya menggunakan --add-tags perintah AWS CLI with the.

Sintaks

```
add-tags --arn=<domain_arn> --tag-list Key=<key>,Value=<value>
```

Parameter	Deskripsi
`--arn`	Nama sumber daya Amazon untuk domain OpenSearch Layanan tempat tag dilampirkan.
`--tag-list`	Kumpulan pasangan nilai kunci yang dipisahkan spasi dalam format berikut: `Key=<key>,Value=<value>`

Contoh

Contoh berikut membuat dua tanda untuk domain log:

```
aws opensearch add-tags --arn arn:aws:es:us-east-1:379931976431:domain/logs --tag-list
 Key=service,Value=OpenSearch Key=instances,Value=m3.2xlarge
```

Anda dapat menghapus tag dari domain OpenSearch Layanan menggunakan --remove-tags perintah.

Sintaks

```
remove-tags --arn=<domain_arn> --tag-keys Key=<key>,Value=<value>
```

Parameter	Deskripsi
`--arn`	Nama Sumber Daya Amazon (ARN) untuk domain OpenSearch Layanan tempat tag dilampirkan.
`--tag-keys`	Set pasangan nilai kunci yang dipisahkan spasi yang ingin Anda hapus dari domain Layanan. OpenSearch

Contoh

Contoh berikut menghapus dua tanda dari domain log yang dibuat dalam contoh sebelumnya:

```
aws opensearch remove-tags --arn arn:aws:es:us-east-1:379931976431:domain/logs --tag-
keys service instances
```

Anda dapat melihat tag yang ada untuk domain OpenSearch Layanan dengan --list-tags perintah:

Sintaks

```
list-tags --arn=<domain_arn>
```

Parameter	Deskripsi
`--arn`	Nama Sumber Daya Amazon (ARN) untuk domain OpenSearch Layanan tempat tag dilampirkan.

Contoh

Contoh berikut mencantumkan semua tanda sumber daya untuk domain log:

```
aws opensearch list-tags --arn arn:aws:es:us-east-1:379931976431:domain/logs
```

Bekerja dengan tag (AWS SDK)

AWS SDK (kecuali SDK Android dan iOS) mendukung semua tindakan yang ditentukan dalam Referensi API OpenSearch Layanan Amazon, termasuk, `AddTagsListTags`, dan `RemoveTags` operasi. Untuk informasi selengkapnya tentang menginstal dan menggunakan AWS SDK, lihat Kit PengembanganAWS Perangkat Lunak.

Python

Contoh ini menggunakan klien Python OpenSearchServicetingkat rendah dari AWS SDK for Python (Boto) untuk menambahkan tag ke domain, mencantumkan tag yang dilampirkan ke domain, dan menghapus tag dari domain. Anda harus memberikan nilai untuk `DOMAIN_ARN`, `TAG_KEY`, dan `TAG_VALUE`.

```
import boto3
from botocore.config import Config  # import configuration

DOMAIN_ARN = ''  # ARN for the domain. i.e "arn:aws:es:us-east-1:123456789012:domain/
my-domain
TAG_KEY = ''  # The name of the tag key. i.e 'Smileyface'
TAG_VALUE = ''  # The value assigned to the tag. i.e 'Practicetag'

# defines the configurations parameters such as region
```

```
my_config = Config(region_name='us-east-1')
client = boto3.client('opensearch', config=my_config)


# defines the client variable

def addTags():
    """Adds tags to the domain"""

    response = client.add_tags(ARN=DOMAIN_ARN,
                               TagList=[{'Key': TAG_KEY,
                                         'Value': TAG_VALUE}])

    print(response)


def listTags():
    """List tags that have been added to the domain"""

    response = client.list_tags(ARN=DOMAIN_ARN)
    print(response)


def removeTags():
    """Remove tags that have been added to the domain"""

    response = client.remove_tags(ARN=DOMAIN_ARN, TagKeys=[TAG_KEY])

    print('Tag removed')
    return response
```

Melakukan tindakan administratif pada domain OpenSearch Layanan Amazon

Amazon OpenSearch Service menawarkan beberapa opsi administratif yang menyediakan kontrol terperinci jika Anda perlu memecahkan masalah dengan domain Anda. Pilihan ini termasuk kemampuan untuk memulai kembali OpenSearch proses pada node data dan kemampuan untuk me-restart node data.

OpenSearch Layanan memantau parameter kesehatan simpul dan, ketika ada anomoli, mengambil tindakan korektif untuk menjaga domain tetap stabil. Dengan opsi administratif untuk memulai ulang

OpenSearch proses pada node, dan memulai ulang node itu sendiri, Anda memiliki kendali atas beberapa tindakan mitigasi ini.

Anda dapat menggunakanAWS Management Console,AWS CLI, atau AWS SDK untuk melakukan tindakan ini. Bagian berikut mencakup cara melakukan tindakan ini dengan konsol.

Mulai ulang OpenSearch proses pada node

Untuk memulai ulang OpenSearch proses pada node

1. Arahkan ke konsol OpenSearch Layanan dihttps://console.aws.amazon.com/aos/.
2. Di panel navigasi kiri, pilih Domain. Pilih nama domain yang ingin Anda gunakan.
3. Setelah halaman detail domain terbuka, navigasikan ke tab Kesehatan instans.
4. Di bawah Node data, pilih tombol di sebelah simpul yang ingin Anda mulai ulang prosesnya.
5. Pilih dropdown Actions dan pilih Restart OpenSearch /Elasticsearch process.
6. Pilih Konfirmasi pada modal.
7. Untuk melihat status tindakan yang Anda mulai, pilih nama node. Setelah halaman rincian node terbuka, pilih tab Events di bawah nama node untuk melihat daftar peristiwa yang terkait dengan node tersebut.

Nyalakan ulang simpul data

Untuk me-reboot node data

1. Arahkan ke konsol OpenSearch Layanan dihttps://console.aws.amazon.com/aos/.
2. Di panel navigasi kiri, pilih Domain. Pilih nama domain yang ingin Anda gunakan.
3. Setelah halaman detail domain terbuka, navigasikan ke tab Kesehatan instans.
4. Di bawah Node data, pilih tombol di sebelah simpul yang ingin Anda mulai ulang prosesnya.
5. Pilih dropdown Actions dan pilih Reboot node.
6. Pilih Konfirmasi pada modal.
7. Untuk melihat status tindakan yang Anda mulai, pilih nama node. Setelah halaman rincian node terbuka, pilih tab Events di bawah nama node untuk melihat daftar peristiwa yang terkait dengan node tersebut.

Mulai ulang proses Dashboard atau Kibana pada node

Untuk memulai ulang proses Dashboard atau Kibana pada node

1. Arahkan ke konsol OpenSearch Layanan dihttps://console.aws.amazon.com/aos/.
2. Di panel navigasi kiri, pilih Domain. Pilih nama domain yang ingin Anda gunakan.
3. Setelah halaman detail domain terbuka, navigasikan ke tab Kesehatan instans.
4. Di bawah Node data, pilih tombol di sebelah simpul yang ingin Anda mulai ulang prosesnya.
5. Pilih dropdown Actions dan pilih Restart Dashboard/Proses Kibana.
6. Pilih Konfirmasi pada modal.
7. Untuk melihat status tindakan yang Anda mulai, pilih nama node. Setelah halaman rincian node terbuka, pilih tab Events di bawah nama node untuk melihat daftar peristiwa yang terkait dengan node tersebut.

Batasan

Opsi administratif memiliki batasan berikut:

- Opsi administratif didukung pada Elasticsearch versi 7.x dan yang lebih tinggi.
- Opsi administratif tidak mendukung domain dengan Multi-AZ dengan Siaga diaktifkan.
- Proses restart OpenSearch dan Elasticsearch didukung pada domain dengan tiga atau lebih node data.
- Dukungan proses Dasbor dan Kibana didukung pada domain dengan dua atau lebih node data.
- Untuk memulai ulang OpenSearch proses pada node atau reboot node, domain tidak boleh dalam keadaan merah dan semua indeks harus memiliki replika yang dikonfigurasi.

Bekerja dengan kueri langsung OpenSearch Layanan Amazon dengan Amazon S3 (pratinjau)

⚠ Ini adalah dokumentasi prarilis untuk kueri langsung OpenSearch Layanan Amazon dengan Amazon S3, yang dalam rilis pratinjau. Dokumentasi dan fitur dapat berubah. Sebaiknya gunakan fitur ini hanya dalam lingkungan pengujian, bukan dalam lingkungan produksi. Untuk syarat dan ketentuan pratinjau, lihat Beta dan Pratinjau di Persyaratan LayananAWS.

Anda dapat menggunakan kueri langsung Amazon OpenSearch Service untuk menanyakan data di Amazon S3. Amazon OpenSearch Service menyediakan integrasi kueri langsung dengan Amazon S3 sebagai cara untuk menganalisis log operasional di Amazon S3 dan data lake yang berbasis di Amazon S3 tanpa harus beralih antar layanan. Anda sekarang dapat menganalisis data di toko objek cloud—dan secara bersamaan menggunakan analisis operasional dan visualisasi Layanan. OpenSearch

Dengan kueri langsung dengan Amazon S3, Anda tidak perlu lagi membangun pipeline ETL yang kompleks atau mengeluarkan biaya duplikasi data di penyimpanan Layanan OpenSearch dan Amazon S3. Anda juga dapat menginstal integrasi template tipe log populer yang menyertakan dasbor yang telah ditentukan sebelumnya, dan mengonfigurasi akselerasi data yang disesuaikan dengan jenis log tersebut. Template termasuk Log Aliran VPC, log, dan AWS CloudTrail log Amazon S3. Akselerasi termasuk melewatkan indeks, tampilan terwujud, dan indeks tertutup.

Topik

Harga

Anda membayar OpenSearch layanan yang ada dan sumber daya Amazon S3 yang digunakan untuk membuat dan memproses kueri langsung. Kueri yang dikirim ke Amazon S3 menggunakan komputasi yang dapat ditagih dan muncul OpenSearch sebagai Unit Komputasi (OCU) per jam.

Kueri langsung dengan Amazon S3 terdiri dari dua jenis — pemeliharaan interaktif dan indeks. Kueri interaktif melakukan analitik pada data Anda di Amazon S3. Ketika Anda menjalankan kueri baru, OpenSearch Layanan memulai sesi baru yang berlangsung selama minimal sepuluh menit. OpenSearch Layanan membuat sesi tetap aktif untuk memastikan bahwa kueri berikutnya berjalan dengan cepat. Kueri pemeliharaan indeks menggunakan komputasi untuk mempertahankan indeks di Layanan. OpenSearch Kueri ini biasanya memakan waktu lebih lama karena mereka menyerap sejumlah data yang dapat dikonfigurasi ke dalam OpenSearch Layanan untuk membuat kueri interaktif berjalan lebih cepat.

Untuk informasi selengkapnya, lihat Harga OpenSearch Layanan Amazon.

Batasan

Batasan berikut berlaku untuk kueri langsung OpenSearch Layanan dengan Amazon S3.

- OpenSearch Domain Anda harus versi 2.11 atau yang lebih baru untuk mendukung kueri langsung OpenSearch Layanan.
- OpenSearch Kueri langsung layanan dengan Amazon S3 hanya mendukung tabel Spark di dalam file. AWS Glue Data CatalogTabel sarang tidak mendukung streaming Spark, yang diperlukan untuk menjaga indeks tetap up to date.
- Beberapa jenis data tidak didukung. Tipe data yang didukung terbatas pada Parket, CSV, dan JSON.
- AWS CloudFormation template tidak didukung dalam rilis pratinjau kueri langsung.
- OpenSearch Domain Anda dan AWS Glue Data Catalog harus sama Akun AWS. Tabel Amazon S3 Anda dapat berada di akun yang berbeda, tetapi harus Wilayah AWS sama dengan domain Anda.
- Struktur Spark Bersarang tidak didukung. Jika data sumber Anda menggunakan struktur bersarang, Anda harus meledakkannya menjadi baris.
- Tabel yang dibuat melalui Athena tidak didukung.

- Kolom yang hilang mungkin memerlukan penggunaan fungsi `COALESCE` SQL untuk mengembalikan hasil.
- Tidak tersedia di OpenSearch Serverless
- Data harus diratakan sebelum kueri atau Anda harus menggunakan SQL di OpenSearch Layanan untuk mengubah kolom bersarang Anda menjadi kolom khusus.

Kuota

Akun Anda memiliki kuota berikut yang terkait dengan kueri langsung OpenSearch Layanan dengan Amazon S3. Setiap kali Anda memulai kueri, OpenSearch Layanan membuka sesi dan membuatnya tetap hidup setidaknya selama sepuluh menit. Ini mengurangi latensi kueri dengan menghapus waktu startup sesi di kueri berikutnya.

Deskripsi	Maxiumum
Koneksi per domain	20
Sumber data per domain	20
Indeks per domain	50
Sesi bersamaan per sumber data	100

Wilayah yang Didukung

Wilayah berikut tersedia untuk pertanyaan langsung OpenSearch Layanan dengan Amazon S3: Asia Pasifik (Tokyo), Eropa (Frankfurt), Eropa (Irlandia), AS Timur (Virginia N.), AS Timur (Ohio), dan AS Barat (Oregon).

Membuat integrasi sumber data OpenSearch Layanan Amazon dengan Amazon S3

⚠ Ini adalah dokumentasi prarilis untuk kueri langsung OpenSearch Layanan Amazon dengan Amazon S3, yang dalam rilis pratinjau. Dokumentasi dan fitur dapat berubah. Sebaiknya

gunakan fitur ini hanya dalam lingkungan pengujian, bukan dalam lingkungan produksi. Untuk syarat dan ketentuan pratinjau, lihat Beta dan Pratinjau di Persyaratan LayananAWS.

Anda dapat membuat sumber data kueri langsung Amazon S3 baru untuk OpenSearch Layanan melalui atau API. AWS Management Console Setiap sumber data baru menggunakan tabel AWS Glue Data Catalog untuk mengelola yang mewakili bucket Amazon S3.

Topik

Prasyarat

Sebelum Anda membuat sumber data, Anda harus memiliki yang berikut:

- OpenSearch Domain dengan versi 2.11 atau yang lebih baru

Untuk petunjuk untuk mengatur ini, lihat the section called “ Membuat domain OpenSearch Layanan” dan Memulai dengan AWS Glue Data Catalog.

Izin yang diperlukan

Untuk membuat sumber data, pengguna atau peran Anda harus memiliki kebijakan berbasis identitas terlampir dengan izin IAM yang sesuai. Kebijakan contoh berikut menunjukkan izin hak istimewa terkecil yang diperlukan untuk membuat dan mengelola sumber data. Perhatikan bahwa jika Anda memiliki izin yang lebih luas, seperti `s3:*` atau `AdministratorAccess` kebijakan, izin ini mencakup izin hak istimewa terkecil dalam kebijakan sampel.

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

```
                "es:ESHttp*",
                "es:AddDataSource",
                "es:DeleteDataSource",
                "es:GetDataSource",
                "es:ListDataSource",
                "es:UpdateDataSource",
                "s3:Get*",
                "s3:List*",
                "s3:Put*",
                "s3:Describe*",
                "glue:*"
            ],
            "Resource": [
                "arn:aws:s3:::bucket-name",
                "arn:aws:s3:::bucket-name/*",
                "arn:aws:glue:us-east-1:{aws-account-id}:database/*"
            ]
        },
        {
            "Sid": "GlueCreateAndReadDataCatalog",
            "Effect": "Allow",
            "Action": [
                "glue:GetDatabase",
                "glue:CreateDatabase",
                "glue:GetDatabases",
                "glue:CreateTable",
                "glue:GetTable",
                "glue:UpdateTable",
                "glue:DeleteTable",
                "glue:GetTables",
                "glue:GetPartition",
                "glue:GetPartitions",
                "glue:CreatePartition",
                "glue:BatchCreatePartition",
                "glue:GetUserDefinedFunctions"
            ],
            "Resource": "*"
        }
    ]
}
```

Peran juga harus memiliki kebijakan kepercayaan berikut, yang menentukan ID target.

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

Untuk petunjuk cara membuat peran, lihat Membuat peran menggunakan kebijakan kepercayaan khusus.

Jika Anda mengaktifkan kontrol akses berbutir halus, peran kontrol akses OpenSearch berbutir halus baru akan dibuat secara otomatis untuk sumber data Anda. <name of data source>Nama peran kontrol akses berbutir halus baru adalah _. AWSOpenSearchDirectQuery

Secara default, peran memiliki akses ke indeks sumber data kueri langsung saja. Meskipun Anda dapat mengonfigurasi peran untuk membatasi atau memberikan akses ke sumber data Anda, sebaiknya Anda tidak menyesuaikan akses peran ini. Jika Anda menghapus sumber data, peran ini akan dihapus. Ini akan menghapus akses untuk pengguna lain jika mereka dipetakan ke peran tersebut.

Petakan AWS Glue Data Catalog peran (jika kontrol akses berbutir halus diaktifkan setelah membuat sumber data)

Jika Anda telah mengaktifkan kontrol akses halus setelah membuat sumber data, Anda harus memetakan pengguna non-admin ke peran IAM dengan AWS Glue Data Catalog akses untuk menjalankan kueri langsung. Untuk membuat `glue_access` peran backend secara manual yang dapat Anda petakan ke peran IAM, lakukan langkah-langkah berikut:

> **Note**
>
> Indeks digunakan untuk kueri apa pun terhadap sumber data. Pengguna dengan akses baca ke indeks permintaan untuk sumber data tertentu dapat membaca semua kueri terhadap

sumber data tersebut. Pengguna dengan akses baca ke indeks hasil dapat membaca hasil untuk semua kueri terhadap sumber data tersebut.

1. Dari menu utama di OpenSearch Dasbor, pilih Keamanan, Peran, dan Buat peran.
2. Beri nama peran glue_access.
3. Untuk izin Cluster, pilih`indices:data/write/bulk*`,`indices:data/read/scroll`,`indices:data/read/scroll/clear`.
4. Untuk Indeks, masukkan indeks berikut yang ingin Anda berikan kepada pengguna akses peran:
   - `.query_execution_request_`*`<name of data source>`*
   - `query_execution_result_`*`<name of data source>`*
   - `flint_*`
5. Untuk izin Indeks, pilih`indices_all`.
6. Pilih Buat.
7. Pilih Pengguna yang dipetakan, Kelola pemetaan.
8. Di bawah peran Backend, tambahkan ARN AWS Glue peran yang memerlukan izin untuk memanggil domain Anda.

   ```
   arn:aws:iam::account-id:role/role-name
   ```

9. Pilih Peta dan konfirmasikan, peran akan muncul pada Pengguna yang Dipetakan.

Untuk informasi selengkapnya tentang peran pemetaan, lihatthe section called “Memetakan peran untuk pengguna”.

Siapkan sumber data kueri langsung baru

Anda dapat mengatur sumber data kueri langsung pada domain dengan AWS Management Console atau API OpenSearch Layanan.

AWS Management Console

1. Arahkan ke konsol OpenSearch Layanan Amazon dihttps://console.aws.amazon.com/aos/.
2. Di panel navigasi kiri, pilih Domain.

3. Pilih domain yang ingin Anda siapkan sumber data baru. Ini membuka halaman detail domain. Pilih tab Koneksi di bawah detail domain umum dan temukan bagian Kueri langsung.
4. Pilih Buat.
5. Pada halaman pembuatan sumber data, masukkan nama untuk sumber data baru Anda. Di bawah Jenis sumber data, pilih Amazon S3. Pilih peran IAM yang ada yang memiliki batasan untuk apa yang dapat diakses di AWS Glue Data Catalog dan Amazon S3.
6. Pilih Buat. Ini membuka layar detail sumber data dengan URL OpenSearch Dasbor. Anda dapat menavigasi ke URL ini untuk menyelesaikan langkah selanjutnya.

OpenSearch API Layanan

Gunakan operasi AddDataSourceAPI untuk membuat sumber data baru di domain Anda.

```
POST https://es.region.amazonaws.com/2021-01-01/opensearch/domain/domain-name/
dataSource

{
   "DataSourceType": {
        "s3GlueDataCatalog": {
            "RoleArn": "arn:aws:iam::account-id:role/Admin"
        }
    }
   "Description": "data-source-description",
   "Name": "my-data-source"
}
```

Langkah selanjutnya

Setelah Anda membuat sumber data, OpenSearch Layanan memberi Anda URL OpenSearch Dasbor. Anda menggunakan ini untuk mengonfigurasi kontrol akses, menentukan tabel, mengatur dasbor berbasis tipe log untuk jenis log populer, dan menanyakan data Anda.

Mengonfigurasi sumber data Anda di Dasbor OpenSearch

⚠ Ini adalah dokumentasi prarilis untuk kueri langsung OpenSearch Layanan Amazon dengan Amazon S3, yang dalam rilis pratinjau. Dokumentasi dan fitur dapat berubah. Sebaiknya

gunakan fitur ini hanya dalam lingkungan pengujian, bukan dalam lingkungan produksi. Untuk syarat dan ketentuan pratinjau, lihat Beta dan Pratinjau di Persyaratan LayananAWS.

Setelah membuat sumber data, Anda dapat mengonfigurasi pengaturan keamanan, menentukan tabel Amazon S3, atau mengatur pengindeksan data yang dipercepat. Bagian ini memandu Anda melalui berbagai kasus penggunaan dengan sumber data Anda di OpenSearch Dasbor sebelum Anda menanyakan data Anda.

Untuk mengonfigurasi bagian berikut, Anda harus terlebih dahulu menavigasi ke sumber data Anda di OpenSearch Dasbor. Di navigasi sebelah kiri, di bawah Manajemen, pilih Sumber data. Di bawah Kelola sumber data, pilih nama sumber data yang Anda buat di konsol.

Mengatur kontrol akses

Pada halaman detail untuk sumber data Anda, temukan bagian Kontrol akses dan pilih Edit. Jika Anda telah menginstal plugin keamanan, pilih Dibatasi dan pilih grup berbasis peran mana yang ingin Anda berikan dengan akses ke sumber data baru. Anda juga dapat memilih Admin hanya jika Anda hanya ingin administrator memiliki akses ke sumber data.

Important

Perhatikan bahwa indeks digunakan untuk kueri apa pun terhadap sumber data, sehingga pengguna dengan akses baca ke indeks permintaan untuk sumber data tertentu dapat membaca semua kueri terhadap sumber data tersebut, dan pengguna dengan akses baca ke indeks hasil dapat membaca hasil untuk semua kueri terhadap sumber data tersebut.

Tentukan AWS Glue Data Catalog tabel

Kueri langsung dari OpenSearch Layanan ke Amazon S3 menggunakan tabel Spark di dalam file. AWS Glue Data CatalogAnda dapat menggunakan Perayap AWS Glue untuk merayapi data Anda, yang akan membuat tabel untuk Anda. Bergantian, Anda dapat secara manual membuat tabel dari dalam Query Workbench.

Untuk mengelola database dan tabel yang ada di sumber data Anda, atau untuk membuat tabel baru yang ingin Anda gunakan kueri langsung, pilih opsi Tentukan tabel pada halaman detail sumber data. Ini membawa Anda ke halaman plugin Query Workbench.

Untuk menyiapkan tabel dengan data sampel yang dapat Anda jelajahi dan gunakan untuk akselerasi di bagian berikut, jalankan kueri berikut:

```
CREATE EXTERNAL TABLE IF NOT EXISTS datasourcename.gluedatabasename.gluetablename (
   `@timestamp` TIMESTAMP,
    clientip STRING,
    request STRING,
    status INT,
    size INT,
    year INT,
    month INT,
    day INT)
USING json PARTITIONED BY(year, month, day) OPTIONS (path 's3://my-bucket/data/
http_log', compression 'bzip2')
```

Setelah membuat tabel, jalankan kueri berikut untuk memastikan bahwa itu kompatibel dengan kueri langsung:

```
MSCK REPAIR TABLE datasourcename.databasename.tablename
```

Mempercepat kueri Anda

Pada halaman detail untuk sumber data Anda, pilih opsi Percepat Kinerja. Untuk memastikan pengalaman yang cepat dengan data Anda di Amazon S3, ada tiga jenis akselerasi berbeda yang dapat Anda atur untuk mengindeks data ke dalam OpenSearch Layanan—melewatkan indeks, tampilan terwujud, dan mencakup indeks.

Melewatkan indeks

Dengan indeks skipping, Anda hanya dapat mengindeks metadata data yang disimpan di Amazon S3. Saat Anda menanyakan tabel dengan indeks lompatan, perencana kueri mereferensikan indeks dan menulis ulang kueri untuk menemukan data secara efisien, alih-alih memindai semua partisi dan file. Hal ini memungkinkan indeks skipping untuk dengan cepat mempersempit lokasi spesifik dari data yang disimpan.

Saat Anda mengonfigurasi tabel Spark yang akan Anda gunakan dari AWS Glue Data Catalog, OpenSearch Dasbor menanyakan apakah Anda ingin membuat indeks lewati pada tabel Anda. Anda dapat membuat indeks skipping di sana, atau Anda dapat membuatnya dengan kasus penggunaan Accelerate Performance setelah Anda menyelesaikan konfigurasi tabel Anda.

```
CREATE SKIPPING INDEX
ON datasourcename.gluedatabasename.gluetablename
(
    year PARTITION,
    month PARTITION,
    day PARTITION,
    hour PARTITION
)
```

Tampilan terwujud

Dengan tampilan terwujud, Anda dapat menggunakan kueri kompleks, seperti agregasi, untuk mendukung visualisasi Dasbor. Tampilan terwujud menyerap sejumlah kecil data Anda ke dalam penyimpanan OpenSearch Layanan. OpenSearch Layanan kemudian membentuk indeks dari data yang dicerna yang dapat Anda gunakan untuk visualisasi. Anda dapat mengelola indeks tampilan terwujud denganthe section called “Manajemen state indeks”, seperti yang Anda bisa dengan OpenSearch indeks lainnya.

Gunakan kueri berikut untuk membuat tampilan terwujud baru untuk `http_logs` tabel yang Anda buat: the section called “Tentukan AWS Glue Data Catalog tabel”

```
CREATE MATERIALIZED VIEW datasourcename.gluedatabasename.viewname_view
AS
  SELECT
    window.start AS `start.time`,
    COUNT(*) AS count
  FROM datasourcename.gluedatabasename.gluetablename
  WHERE status != 200
  GROUP BY TUMBLE(`@timestamp`, '1 Minutes')
WITH (
  auto_refresh = true,
  refresh_interval = '1 Minutes',
  checkpoint_location = 's3://my-bucket/data/http_log/checkpoint_http_count_view',
  watermark_delay = '10 Minutes'
);
```

Meliputi indeks

Dengan indeks penutup, Anda dapat menelan data dari kolom tertentu dalam tabel. Ini adalah yang paling berkinerja dari tiga jenis pengindeksan. Karena OpenSearch Layanan menyerap semua data

dari kolom yang Anda inginkan, Anda mendapatkan kinerja yang lebih baik dan dapat melakukan analisis lanjutan.

Sama seperti tampilan terwujud, OpenSearch Service membuat indeks baru dari data indeks penutup. Anda dapat menggunakan indeks baru ini untuk visualisasi Dasbor dan fungsionalitas OpenSearch Layanan lainnya, seperti deteksi anomali atau kemampuan geospasial. Anda dapat mengelola indeks tampilan penutup denganthe section called “Manajemen state indeks”, seperti yang Anda bisa dengan OpenSearch indeks lainnya.

Gunakan kueri berikut untuk membuat indeks penutup baru untuk `http_logs` tabel yang Anda buatthe section called “Tentukan AWS Glue Data Catalog tabel”:

```
CREATE INDEX status_clientip_and_day
ON datasourcename.gluedatabasename.gluetablename ( status, day, clientip )
WITH (
  auto_refresh = true,
  refresh_interval = '5 minute',
  checkpoint_location = 's3://my-bucket/data/http_log/checkpoint_status_and_day'
)
```

Kueri data di Dasbor OpenSearch

> ⚠ Ini adalah dokumentasi prarilis untuk kueri langsung OpenSearch Layanan Amazon dengan Amazon S3, yang dalam rilis pratinjau. Dokumentasi dan fitur dapat berubah. Sebaiknya gunakan fitur ini hanya dalam lingkungan pengujian, bukan dalam lingkungan produksi. Untuk syarat dan ketentuan pratinjau, lihat Beta dan Pratinjau di Persyaratan LayananAWS.

Setelah Anda mengatur tabel dan mengonfigurasi akselerasi kueri opsional yang Anda inginkan, Anda sekarang dapat mulai melakukan analitik pada data Anda. Untuk melakukan kueri data, pilih sumber data dari menu tarik-turun di halaman Discover atau halaman Observability di Dasbor. OpenSearch

Jika Anda menggunakan indeks skipping atau belum membuat indeks, Anda dapat menggunakan SQL atau Piped Processing Language (PPL) untuk menanyakan data Anda. Jika Anda telah mengonfigurasi tampilan terwujud atau indeks penutup, Anda sudah memiliki indeks dan dapat menggunakan Dashboards Query Language (DQL) di seluruh Dasbor. Anda juga dapat

menggunakan PPL dengan plugin Observability, dan SQL dengan plugin Query Workbench. Saat ini, hanya plugin Observability dan Query Workbench yang mendukung PPL dan SQL.

SQL

Gunakan kueri berikut untuk menjalankan contoh kueri SQL untuk `http_logs` tabel yang Anda buat: the section called “Tentukan AWS Glue Data Catalog tabel”

```
SELECT
    FIRST(day) AS day,
    status,
    COUNT(status) AS status_count_by_day
FROM datasourcename.gluedatabasename.gluetablename
WHERE status >= 400
GROUP BY day, status
ORDER BY day, status
LIMIT 20;
```

PPL

Gunakan kueri berikut untuk menjalankan contoh kueri PPL untuk `http_logs` tabel yang Anda buat: the section called “Tentukan AWS Glue Data Catalog tabel”

```
source = datasourcename.gluedatabasename.gluetablename |
where status = 500 | sort - clientip, @timestamp | head 20
```

Menghapus sumber data OpenSearch Layanan Amazon dengan Amazon S3

⚠ Ini adalah dokumentasi prarilis untuk kueri langsung OpenSearch Layanan Amazon dengan Amazon S3, yang dalam rilis pratinjau. Dokumentasi dan fitur dapat berubah. Sebaiknya gunakan fitur ini hanya dalam lingkungan pengujian, bukan dalam lingkungan produksi. Untuk syarat dan ketentuan pratinjau, lihat Beta dan Pratinjau di Persyaratan LayananAWS.

Saat Anda menghapus sumber data, Amazon OpenSearch Service menghapusnya dari domain Anda. OpenSearch Layanan juga menghapus indeks yang terkait dengan sumber data. Data

transaksional Anda tidak dihapus dari Amazon S3, tetapi Amazon S3 tidak mengirim data baru ke Layanan. OpenSearch

Anda dapat menghapus integrasi sumber data menggunakan AWS Management Console atau OpenSearch Service API.

AWS Management Console

Untuk menghapus sumber data

1. Arahkan ke konsol OpenSearch Layanan Amazon dihttps://console.aws.amazon.com/aos/.
2. Dari panel navigasi kiri, pilih Domain.
3. Pilih domain yang ingin Anda hapus sumber datanya. Ini membuka halaman detail domain. Pilih tab Koneksi di bawah informasi umum dan temukan bagian Kueri langsung.
4. Pilih sumber data yang ingin Anda hapus, pilih Hapus, dan konfirmasi penghapusan.

OpenSearch API Layanan

Gunakan operasi DeleteDataSourceAPI untuk menghapus sumber data yang ada di domain Anda.

```
POST https://es.region.amazonaws.com/2021-01-01/opensearch/domain/domain-name/
dataSource/data-source-name
```

Memantau domain OpenSearch Layanan Amazon

Pemantauan adalah bagian penting dari pemeliharaan keandalan, ketersediaan, dan performa Amazon OpenSearch Service dan AWS solusi Anda. AWSmenyediakan alat berikut untuk memantau sumber daya OpenSearch layanan, melaporkan masalah, dan mengambil tindakan otomatis jika diperlukan:

Amazon CloudWatch

Amazon CloudWatch memonitor sumber daya OpenSearch layanan Anda secara waktu nyata. Anda dapat mengumpulkan dan melacak metrik, membuat dasbor yang disesuaikan, dan mengatur alarm yang memberi tahu Anda atau mengambil tindakan saat metrik mencapai ambang batas tertentu. Untuk informasi lebih lanjut, lihat Panduan Pengguna Amazon CloudWatch.

CloudWatchLog Amazon

Amazon CloudWatch Logs memungkinkan Anda memantau, menyimpan, dan mengakses file OpenSearch log Anda. CloudWatchLog memantau informasi dalam file log dan memberi tahu Anda ketika ambang tertentu terpenuhi. Untuk informasi selengkapnya, lihat Panduan Pengguna Log CloudWatch Amazon.

Amazon EventBridge

Amazon EventBridge memberikan aliran kejadian sistem secara hampir waktu nyata yang menjelaskan perubahan dalam domain OpenSearch Service Anda. Anda dapat membuat aturan yang mengawasi kejadian tertentu, dan memicu tindakan otomatis dalam layanan AWS lainnya saat tindakan ini terjadi. Untuk informasi lebih lanjut, lihat Panduan Pengguna Amazon EventBridge.

AWS CloudTrail

AWS CloudTrailmenangkap panggilan API konfigurasi yang dibuat ke OpenSearch Service sebagai peristiwa. Dapat mengirimkan peristiwa ini ke bucket Amazon S3 yang Anda tentukan. Dengan menggunakan informasi ini, Anda dapat mengidentifikasi pengguna dan akun mana yang meminta, alamat IP sumber tempat permintaan dibuat, dan kapan permintaan tersebut terjadi. Untuk informasi selengkapnya, lihat AWSPanduan Pengguna CloudTrail.

Topik

- Memantau metrik OpenSearch klaster dengan Amazon CloudWatch
- Memantau OpenSearch log dengan Amazon CloudWatch Logs
- Memantau log audit di Amazon OpenSearch Service
- Memantau peristiwa OpenSearch Layanan dengan Amazon EventBridge
- Pemantauan panggilan API Amazon OpenSearch Service dengan AWS CloudTrail

Memantau metrik OpenSearch klaster dengan Amazon CloudWatch

OpenSearch Layanan Amazon menerbitkan data dari domain Anda ke Amazon. CloudWatch CloudWatch memungkinkan Anda mengambil statistik tentang titik-titik data tersebut sebagai kumpulan data deret waktu yang diurutkan, yang dikenal sebagai metrik. OpenSearch Layanan mengirimkan sebagian besar metrik CloudWatch dalam interval 60 detik. Jika Anda menggunakan volume Tujuan Umum atau EBS Magnetis, metrik volume EBS diperbarui hanya setiap lima menit. Untuk informasi selengkapnya tentang Amazon CloudWatch, lihat Panduan CloudWatch Pengguna Amazon.

Konsol OpenSearch Layanan menampilkan serangkaian bagan berdasarkan data mentah dari CloudWatch. Bergantung pada kebutuhan Anda, Anda mungkin lebih suka melihat data cluster CloudWatch daripada grafik di konsol. Layanan mengarsipkan metrik selama dua minggu sebelum membuangnya. Metrik disediakan tanpa biaya tambahan, tetapi CloudWatch masih dikenakan biaya untuk membuat dasbor dan alarm. Untuk informasi lebih lanjut, lihat harga Amazon CloudWatch.

OpenSearch Layanan menerbitkan metrik berikut ke: CloudWatch

- the section called “Metrik klaster”
- the section called “Metrik simpul utama khusus”
- the section called “Metrik volume EBS”
- the section called “Metrik instans”
- the section called “UltraWarm metrik”
- the section called “Metrik penyimpanan dingin”
- the section called “Metrik pemberitahuan”
- the section called “Metrik deteksi anomali”

- the section called “Metrik pencarian asinkron”
- the section called “Metrik SQL”
- the section called “metrik k-NN”
- the section called “Metrik pencarian lintas klaster”
- the section called “Metrik replikasi lintas-cluster”
- the section called “Metrik Learning to Rank”
- the section called “Metrik Bahasa Pemrosesan yang Disalurkan”

Melihat metrik di CloudWatch

CloudWatch metrik dikelompokkan pertama oleh namespace layanan, dan kemudian oleh berbagai kombinasi dimensi dalam setiap namespace.

Untuk melihat metrik menggunakan konsol CloudWatch

1. Buka CloudWatch konsol di https://console.aws.amazon.com/cloudwatch/.
2. Di panel navigasi kiri, temukan Metrik dan pilih Semua metrik. Pilih ES/ OpenSearchService namespace.
3. Pilih dimensi untuk melihat metrik yang sesuai. Metrik untuk masing-masing simpul berada di dimensi `ClientId, DomainName, NodeId`. Metrik klaster ada di dimensi `Per-Domain, Per-Client Metrics`. Beberapa metrik simpul dikumpulkan di tingkat klaster dan dengan demikian termasuk dalam kedua dimensi. Metrik serpihan berada di dimensi `ClientId, DomainName, NodeId, ShardRole`.

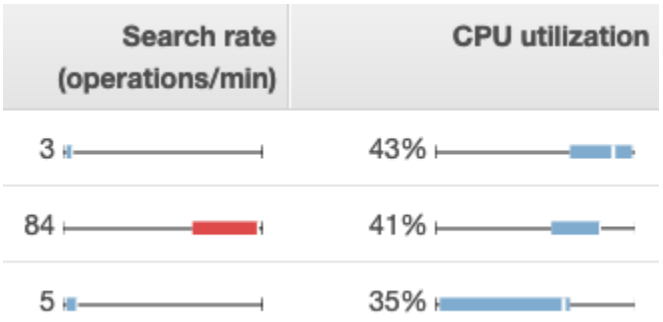
Untuk melihat daftar metrik menggunakan AWS CLI

Jalankan perintah berikut:

```
aws cloudwatch list-metrics --namespace "AWS/ES"
```

Menafsirkan grafik kesehatan dalam Layanan OpenSearch

Untuk melihat metrik di OpenSearch Layanan, gunakan tab kesehatan Cluster dan kesehatan Instance. Tab Instance health menggunakan bagan kotak untuk memberikan at-a-glance visibilitas ke kesehatan setiap OpenSearch node:



- Setiap kotak berwarna menunjukkan rentang nilai untuk simpul selama periode waktu yang ditentukan.
- Kotak biru mewakili nilai-nilai yang konsisten dengan simpul lainnya. Kotak merah mewakili outlier.
- Garis putih dalam setiap kotak menunjukkan nilai simpul saat ini.
- “Whisker” di kedua sisi setiap kotak menunjukkan nilai minimum dan maksimum untuk semua simpul selama periode waktu.

Jika Anda membuat perubahan konfigurasi ke domain Anda, daftar masing-masing instans di tab Kesehatan klaster dan Kesehatan instans sering kali berlipat ganda untuk jangka waktu singkat sebelum kembali ke nomor yang benar. Untuk penjelasan tentang perilaku ini, lihat the section called “Perubahan konfigurasi”.

Metrik klaster

Amazon OpenSearch Service menyediakan metrik berikut untuk cluster.

Metrik	Deskripsi
`ClusterStatus.green`	Nilai 1 menunjukkan bahwa semua serpihan indeks dialokasikan untuk simpul dalam klaster. Statistik yang relevan: Maksimum
`ClusterStatus.yellow`	Nilai 1 menunjukkan bahwa pecahan utama untuk semua indeks dialokasikan ke node di cluster, tetapi pecahan replika untuk setidaknya satu indeks tidak. Untuk informasi selengkapnya, lihat the section called “Status klaster kuning”. Statistik yang relevan: Maksimum

Metrik	Deskripsi
`ClusterStatus.red`	Nilai 1 menunjukkan bahwa serpihan primer dan replika untuk setidaknya satu indeks tidak dialokasikan untuk simpul dalam klaster. Untuk informasi selengkapnya, lihat the section called “Status klaster merah”. Statistik yang relevan: Maksimum
`Shards.active`	Jumlah total aktif serpihan primer dan replika aktif. Statistik yang relevan: Maksimum, Jumlah
`Shards.unassigned`	Jumlah serpihan yang tidak dialokasikan ke simpul di klaster. Statistik yang relevan: Maksimum, Jumlah
`Shards.delayedUnassigned`	Jumlah serpihan yang alokasi simpulnya telah tertunda oleh pengaturan batas waktu. Statistik yang relevan: Maksimum, Jumlah
`Shards.activePrimary`	Jumlah serpihan primer aktif. Statistik yang relevan: Maksimum, Jumlah
`Shards.initializing`	Jumlah serpihan yang berada di bawah inisialisasi. Statistik yang relevan: Jumlah
`Shards.relocating`	Jumlah serpihan yang berada di bawah relokasi. Statistik yang relevan: Jumlah
`Nodes`	Jumlah node di kluster OpenSearch Service, termasuk node master dan UltraWarm node khusus. Untuk informasi selengkapnya, lihat the section called “Perubahan konfigurasi”. Statistik yang relevan: Maksimum

Metrik	Deskripsi
SearchableDocuments	Jumlah total dokumen yang dapat dicari di semua simpul data pada klaster. Statistik yang relevan: Minimum, Maksimum, Rata-rata
DeletedDocuments	Jumlah total dokumen yang ditandai untuk penghapusan di semua simpul data pada klaster. Dokumen-dokumen ini tidak lagi muncul di hasil pencarian, tetapi OpenSearch hanya menghapus dokumen yang dihapus dari disk selama penggabungan segmen. Metrik ini meningkat setelah permintaan hapus dan menurun setelah gabungan segmen. Statistik yang relevan: Minimum, Maksimum, Rata-rata
CPUUtilization	Persentase penggunaan CPU untuk simpul data di klaster. Maksimum menunjukkan simpul dengan penggunaan CPU tertinggi. Rata-rata mewakili semua simpul dalam klaster. Metrik ini juga tersedia untuk masing-masing simpul. Statistik yang relevan: Maksimum, Rata-rata

Metrik	Deskripsi
`FreeStorageSpace`	Ruang kosong untuk simpul data dalam klaster. `Sum` menunjukkan total ruang kosong untuk klaster, tetapi Anda harus meninggalkan periode pada satu menit untuk mendapatkan nilai yang akurat. `Minimum` dan `Maximum` menunjukkan simpul dengan ruang paling sedikit dan paling bebas, menurut urutannya. Metrik ini juga tersedia untuk masing-masing node. OpenSearch Layanan melempar a `ClusterBlockException` saat metrik ini mencapai`0`. Untuk memulihkan, Anda harus menghapus indeks, menambahkan instance yang lebih besar, atau menambahkan penyimpanan berbasis EBS ke instance yang ada. Untuk mempelajari selengkapnya, lihat the section called “Kurangnya ruang penyimpanan yang tersedia”. Konsol OpenSearch Layanan menampilkan nilai ini di GiB. CloudWatch Konsol Amazon menampilkannya di MiB. **Note** `FreeStorageSpace` akan selalu lebih rendah dari nilai yang disediakan `_cat/allocation` API OpenSearch `_cluster/stats` dan. OpenSearch Layanan mencadangkan persentase ruang penyimpanan pada setiap instance untuk operasi internal. Untuk informasi lebih lanjut, lihat Menghitung persyaratan penyimpanan. Statistik yang relevan: Minimum, Maksimum, Rata-rata, Jumlah
`ClusterUsedSpace`	Total penggunaan ruang untuk klaster. Anda harus menyediakan periode pada satu menit untuk mendapatkan nilai yang akurat. Konsol OpenSearch Layanan menampilkan nilai ini di GiB. CloudWatch Konsol Amazon menampilkannya di MiB. Statistik yang relevan: Minimum, Maksimum

Metrik	Deskripsi
`ClusterIndexWritesBlocked`	Menunjukkan apakah klaster Anda menerima atau memblokir permintaan tulis yang masuk. Nilai 0 berarti klaster menerima permintaan. Nilai 1 berarti permintaan diblokir. Beberapa faktor umum mencakup hal berikut ini: `FreeStorageSpace` terlalu rendah atau `JVMMemoryPressure` terlalu tinggi. Untuk mengatasi masalah ini, pertimbangkan untuk menambahkan lebih banyak ruang disk atau menyesuaikan skala klaster Anda. Statistik yang relevan: Maksimum
`JVMMemoryPressure`	Persentase maksimum heap Java yang digunakan untuk semua node data di cluster. OpenSearch Layanan menggunakan setengah dari RAM instance untuk heap Java, hingga ukuran heap 32 GiB. Anda dapat menskalakan instans secara vertikal hingga 64 GiB RAM, di mana Anda dapat menskalakan secara horizontal dengan menambahkan instans. Lihat the section called “ CloudWatch Alarm yang direkomendasikan”. Statistik yang relevan: Maksimum **Note** Logika untuk metrik ini berubah dalam perangkat lunak layanan R20220323. Untuk informasi selengkapnya, lihat catatan rilis.
`OldGenJVMMemoryPressure`	Persentase maksimum heap Java yang digunakan untuk “generasi lama” pada semua node data di cluster. Metrik ini juga tersedia di tingkat node. Statistik yang relevan: Maksimum

Metrik	Deskripsi
AutomatedSnapshotFailure	Jumlah snapshot otomatis yang gagal untuk klaster. Nilai 1 menunjukkan ketiadaan snapshot otomatis yang diambil untuk domain dalam 36 jam sebelumnya. Statistik yang relevan: Minimum, Maksimum
CPUCreditBalance	Sisa kredit CPU yang tersedia untuk simpul data dalam klaster. Kredit CPU memberikan performa inti CPU penuh selama satu menit. Untuk informasi selengkapnya, lihat Kredit CPU di Panduan Developer Amazon EC2. Metrik ini hanya tersedia untuk tipe instans T2. Statistik yang relevan: Minimum
OpenSearchDashboardsHealthyNodes	Pemeriksaan kesehatan untuk OpenSearch Dasbor. Jika minimum, maksimum, dan rata-rata semuanya sama dengan 1, Dasbor berperilaku normal. Jika Anda memiliki 10 simpul dengan maksimum 1, minimal 0, dan rata-rata 0,7, ini berarti 7 simpul (70%) sehat dan 3 simpul (30%) tidak sehat. Statistik yang relevan: Minimum, Maksimum, Rata-rata
OpensearchDashboardsReportingFailedRequestSysErrCount	Jumlah permintaan untuk menghasilkan laporan OpenSearch Dasbor yang gagal karena masalah server atau keterbatasan fitur. Statistik yang relevan: Jumlah
OpensearchDashboardsReportingFailedRequestUserErrCount	Jumlah permintaan untuk menghasilkan laporan OpenSearch Dasbor yang gagal karena masalah klien. Statistik yang relevan: Jumlah
OpensearchDashboardsReportingRequestCount	Jumlah total permintaan untuk menghasilkan laporan OpenSearch Dasbor. Statistik yang relevan: Jumlah

Metrik	Deskripsi
`OpensearchDashboardsReportingSuccessCount`	Jumlah permintaan yang berhasil untuk menghasilkan laporan OpenSearch Dasbor. Statistik yang relevan: Jumlah
`KMSKeyError`	Nilai 1 menunjukkan bahwa AWS KMS kunci yang digunakan untuk mengenkripsi data saat istirahat telah dinonaktifkan. Untuk memulihkan domain untuk operasi normal, aktifkan kembali kunci. Konsol menampilkan metrik ini hanya untuk domain yang mengenkripsi data tidak aktif. Statistik yang relevan: Minimum, Maksimum
`KMSKeyInaccessible`	Nilai 1 menunjukkan bahwa AWS KMS kunci yang digunakan untuk mengenkripsi data saat istirahat telah dihapus atau dicabut hibahnya ke Layanan. OpenSearch Anda tidak dapat memulihkan domain yang berada dalam keadaan ini. Jika Anda memiliki snapshot manual, Anda dapat menggunakannya untuk memigrasi data domain ke domain baru. Konsol menampilkan metrik ini hanya untuk domain yang mengenkripsi data tidak aktif. Statistik yang relevan: Minimum, Maksimum
`InvalidHostHeaderRequests`	Jumlah permintaan HTTP yang dibuat ke OpenSearch cluster yang menyertakan header host yang tidak valid (atau hilang). Permintaan yang valid menyertakan nama host domain sebagai nilai header host. OpenSearch Layanan menolak permintaan yang tidak valid untuk domain akses publik yang tidak memiliki kebijakan akses terbatas. Anda sebaiknya menerapkan kebijakan akses terbatas ke semua domain. Jika Anda melihat nilai besar untuk metrik ini, konfirmasikan bahwa OpenSearch klien Anda menyertakan nama host domain (dan bukan, misalnya, alamat IP-nya) dalam permintaan mereka. Statistik yang relevan: Jumlah

Metrik	Deskripsi
`OpenSearchRequests (previously ElasticsearchRequests)`	Jumlah permintaan yang dibuat ke OpenSearch cluster. Statistik yang relevan: Jumlah
`2xx, 3xx, 4xx, 5xx`	Jumlah permintaan ke domain yang menghasilkan kode respon HTTP yang diberikan (2xx, 3xx, 4xx, 5xx). Statistik yang relevan: Jumlah
`ThroughputThrottle`	Menunjukkan apakah disk telah dibatasi atau tidak. Throttling terjadi ketika throughput gabungan `ReadThroughputMicroBursting` dan `WriteThroughputMicroBursting` lebih tinggi dari throughput maksimum,. `MaxProvisionedThroughput` `MaxProvisionedThroughput` adalah nilai yang lebih rendah dari throughput instance atau throughput volume yang disediakan. Nilai 1 menunjukkan bahwa disk telah dibatasi. Nilai 0 menunjukkan perilaku normal. Untuk informasi tentang throughput instans, lihat Instans yang dioptimalkan Amazon EBS. Untuk informasi tentang throughput volume, lihat Jenis volume Amazon EBS. Statistik yang relevan: Minimum, Maksimum

Metrik simpul utama khusus

Amazon OpenSearch Service menyediakan metrik berikut untuk node master khusus.

Metrik	Deskripsi
`MasterCPUUtilization`	Persentase maksimum sumber daya CPU yang digunakan oleh simpul utama khusus. Sebaiknya tingkatkan ukuran tipe instans saat metrik ini mencapai 60 persen. Statistik yang relevan: Maksimum

Metrik	Deskripsi
`MasterFreeStorageSpace`	Metrik ini tidak relevan dan bisa diabaikan. Layanan tidak menggunakan simpul utama sebagai data simpul.
`MasterJVMMemoryPressure`	Persentase maksimum tumpukan Java yang digunakan untuk semua simpul utama khusus di klaster. Sebaiknya lakukan pemindahan ke tipe instans yang lebih besar bila metrik ini mencapai 85 persen. Statistik yang relevan: Maksimum **Note** Logika untuk metrik ini berubah dalam perangkat lunak layanan R20220323. Untuk informasi selengkapnya, lihat catatan rilis.
`MasterOldGenJVMMemoryPressure`	Persentase maksimum heap Java yang digunakan untuk "generasi lama" per master node. Statistik yang relevan: Maksimum
`MasterCPUCreditBalance`	Kredit CPU yang tersisa tersedia untuk simpul utama khusus dalam klaster. Kredit CPU memberikan performa inti CPU penuh selama satu menit. Untuk informasi selengkapnya, lihat Kredit CPU di Panduan Developer Amazon EC2. Metrik ini hanya tersedia untuk tipe instans T2. Statistik yang relevan: Minimum

Metrik	Deskripsi
`MasterReachableFromNode`	Pemeriksaan kondisi untuk pengecualian `MasterNotDiscovered` . Nilai 1 menunjukkan perilaku normal. Nilai 0 menunjukkan bahwa `/_cluster/health/` gagal. Kegagalan berarti bahwa node master tidak dapat dijangkau dari node sumber. Mereka biasanya hasil dari masalah konektivitas jaringan atau masalah AWS ketergantungan. Statistik yang relevan: Maksimum
`MasterSysMemoryUtilization`	Persentase memori simpul utama yang sedang digunakan. Statistik yang relevan: Maksimum

Metrik volume EBS

Amazon OpenSearch Service menyediakan metrik berikut untuk volume EBS.

Metrik	Deskripsi
`ReadLatency`	Latensi, dalam hitungan detik, untuk operasi baca pada volume EBS. Metrik ini juga tersedia untuk masing-masing simpul. Statistik yang relevan: Minimum, Maksimum, Rata-rata
`WriteLatency`	Latensi, dalam hitungan detik, untuk operasi tulis pada volume EBS. Metrik ini juga tersedia untuk masing-masing simpul. Statistik yang relevan: Minimum, Maksimum, Rata-rata
`ReadThroughput`	Throughput, dalam byte per detik, untuk operasi baca pada volume EBS. Metrik ini juga tersedia untuk masing-masing simpul. Statistik yang relevan: Minimum, Maksimum, Rata-rata

Metrik	Deskripsi
`ReadThroughputMicroBursting`	Throughput, dalam byte per detik, untuk operasi baca pada volume EBS saat ledakan mikro dipertimbangkan. Metrik ini juga tersedia untuk masing-masing simpul. Micro-bursting terjadi ketika volume EBS meledak IOPS tinggi atau throughput untuk periode waktu yang jauh lebih pendek (kurang dari satu menit). Statistik yang relevan: Minimum, Maksimum, Rata-rata
`WriteThroughput`	Throughput, dalam byte per detik, untuk operasi tulis pada volume EBS. Metrik ini juga tersedia untuk masing-masing simpul. Statistik yang relevan: Minimum, Maksimum, Rata-rata
`WriteThroughputMicroBursting`	Throughput, dalam byte per detik, untuk operasi penulisan pada volume EBS saat micro-bursting dipertimbangkan. Metrik ini juga tersedia untuk masing-masing simpul. Micro-bursting terjadi ketika volume EBS meledak IOPS tinggi atau throughput untuk periode waktu yang jauh lebih pendek (kurang dari satu menit). Statistik yang relevan: Minimum, Maksimum, Rata-rata
`DiskQueueDepth`	Jumlah permintaan input dan output (I/O) yang tertunda untuk volume EBS. Statistik yang relevan: Minimum, Maksimum, Rata-rata
`ReadIOPS`	Jumlah operasi input dan output (I/O) per detik untuk operasi baca pada volume EBS. Metrik ini juga tersedia untuk masing-masing simpul. Statistik yang relevan: Minimum, Maksimum, Rata-rata
`ReadIOPSMicroBursting`	Jumlah operasi input dan output (I/O) per detik untuk operasi baca pada volume EBS saat micro-bursting dipertimbangkan. Metrik ini juga tersedia untuk masing-masing simpul. Micro-bursting terjadi ketika volume EBS meledak IOPS tinggi atau throughput untuk periode waktu yang jauh lebih pendek (kurang dari satu menit). Statistik yang relevan: Minimum, Maksimum, Rata-rata

Metrik	Deskripsi
`WriteIOPS`	Jumlah operasi input dan output (I/O) per detik untuk operasi tulis pada volume EBS. Metrik ini juga tersedia untuk masing-masing simpul. Statistik yang relevan: Minimum, Maksimum, Rata-rata
`WriteIOPS MicroBursting`	Jumlah operasi input dan output (I/O) per detik untuk operasi tulis pada volume EBS saat micro-bursting dipertimbangkan. Metrik ini juga tersedia untuk masing-masing simpul. Micro-bursting terjadi ketika volume EBS meledak IOPS tinggi atau throughput untuk periode waktu yang jauh lebih pendek (kurang dari satu menit). Statistik yang relevan: Minimum, Maksimum, Rata-rata
`BurstBalance`	Persentase kredit input dan output (I/O) yang tersisa di bucket burst untuk volume EBS. Nilai 100 berarti bahwa volume telah mengumpul kan jumlah kredit maksimum. Jika persentase ini turun di bawah 70%, lihatthe section called “Keseimbangan burst EBS rendah”. Saldo burst tetap pada 0 untuk domain dengan tipe volume gp3, dan domain dengan volume gp2 yang memiliki ukuran volume di atas 1000 GiB. Statistik yang relevan: Minimum, Maksimum, Rata-rata

Metrik instans

Amazon OpenSearch Service menyediakan metrik berikut untuk setiap instans dalam domain. OpenSearch Layanan juga menggabungkan metrik instans ini untuk memberikan wawasan tentang kesehatan klaster secara keseluruhan. Anda dapat memverifikasi perilaku ini dengan menggunakan statistik Jumlah Sampel dalam konsol. Perhatikan bahwa setiap metrik dalam tabel berikut memiliki statistik yang relevan untuk simpul dan klaster.

> **Important**
>
> Versi yang berbeda dari Elasticsearch menggunakan kolam utas yang berbeda untuk memproses panggilan ke API `_index`. Elasticsearch 1.5 dan 2.3 menggunakan kolam utas indeks. Elasticsearch 5. x, 6.0, dan 6.2 menggunakan kumpulan utas massal. OpenSearch

dan Elasticsearch 6.3 dan yang lebih baru gunakan kumpulan utas tulis. Saat ini, konsol OpenSearch Layanan tidak menyertakan grafik untuk kumpulan utas massal. Gunakan `GET _cluster/settings?include_defaults=true` untuk memeriksa kolam utas dan ukuran antrean untuk klaster Anda.

Metrik	Deskripsi
`IndexingLatency`	Perbedaan total waktu, dalam milidetik, diambil oleh semua operasi pengindeksan dalam simpul antara menit N dan menit (N-1). Statistik simpul yang relevan: Rata-rata Statistik klaster yang relevan: Rata-rata, Maksimum
`IndexingRate`	Jumlah operasi pengindeksan per menit. Satu panggilan ke API `_bulk` yang menambahkan dua dokumen dan memperbarui dua dianggap sebagai empat operasi, yang mungkin tersebar di satu atau beberapa simpul. Jika indeks yang memiliki satu atau beberapa replika, simpul lain dalam klaster juga mencatat total empat operasi pengindeksan. Penghapusan dokumen tidak dihitung dalam metrik ini. Statistik simpul yang relevan: Rata-rata Statistik klaster yang relevan: Rata-rata, Maksimum, Sum
`SearchLatency`	Perbedaan total waktu, dalam milidetik, diambil oleh semua pencarian dalam simpul antara menit N dan menit (N-1). Statistik simpul yang relevan: Rata-rata Statistik klaster yang relevan: Rata-rata, Maksimum
`SearchRate`	Jumlah total permintaan pencarian per menit untuk semua serpihan pada simpul data. Satu panggilan ke API `_search` mungkin mengembalikan hasilnya dari banyak serpihan yang berbeda. Jika lima serpihan ini berada pada satu simpul, simpul tersebut akan

Metrik	Deskripsi
	melaporkan 5 untuk metrik ini, meskipun klien hanya membuat satu permintaan. Statistik simpul yang relevan: Rata-rata Statistik klaster yang relevan: Rata-rata, Maksimum, Sum
`SegmentCount`	Jumlah segmen pada simpul data. Semakin banyak segmen yang Anda miliki, semakin lama setiap pencarian berlangsung. OpenSearch kadang-kadang menggabungkan segmen yang lebih kecil menjadi yang lebih besar. Statistik simpul yang relevan: Maksimum, Rata-rata Statistik klaster yang relevan: Jumlah, Maksimum, Rata-rata
`SysMemoryUtilization`	Persentase memori instans yang sedang digunakan. Nilai tinggi untuk metrik ini normal dan biasanya tidak mewakili masalah dengan klaster Anda. Untuk indikator yang lebih baik mengenai potensi masalah performa dan stabilitas, lihat metrik `JVMMemoryPressure` . Statistik simpul yang relevan: Minimum, Maksimum, Rata-rata Statistik klaster yang relevan: Minimum, Maksimum, Rata-rata
`JVMGCYoungCollectionCount`	Frekuensi pengumpulan sampah "generasi muda" telah berjalan. Jumlah eksekusi yang besar dan terus bertambah adalah bagian normal dari operasi klaster. Statistik simpul yang relevan: Maksimum Statistik klaster yang relevan: Jumlah, Maksimum, Rata-rata

Metrik	Deskripsi
JVMGCYoungCollectionTime	Jumlah waktu, dalam milidetik, yang telah klaster habiskan untuk melakukan pengumpulan sampah "generasi muda". Statistik simpul yang relevan: Maksimum Statistik klaster yang relevan: Jumlah, Maksimum, Rata-rata
JVMGCOldCollectionCount	Frekuensi pengumpulan sampah "generasi tua" telah berjalan. Dalam sebuah klaster dengan sumber daya yang cukup, jumlah ini harus tetap kecil dan jarang bertumbuh. Statistik simpul yang relevan: Maksimum Statistik klaster yang relevan: Jumlah, Maksimum, Rata-rata
JVMGCOldCollectionTime	Jumlah waktu, dalam milidetik, yang telah klaster habiskan untuk melakukan pengumpulan sampah "generasi tua". Statistik simpul yang relevan: Maksimum Statistik klaster yang relevan: Jumlah, Maksimum, Rata-rata
OpenSearchDashboardsConcurrentConnections	Jumlah koneksi konkuren aktif ke OpenSearch Dasbor. Jika nomor ini selalu tinggi, pertimbangkan untuk menyesuaikan skala klaster Anda. Statistik simpul yang relevan: Maksimum Statistik klaster yang relevan: Jumlah, Maksimum, Rata-rata
OpenSearchDashboardsHealthyNode	Pemeriksaan kesehatan untuk node OpenSearch Dasbor individu. Nilai 1 menunjukkan perilaku normal. Nilai 0 menunjukkan bahwa Dasbor tidak dapat diakses. Statistik simpul yang relevan: Minimum Statistik klaster yang relevan: Minimum, Maksimum, Rata-rata

Metrik	Deskripsi
OpenSearchDashboardsHeapTotal	Jumlah memori heap yang dialokasikan ke OpenSearch Dasbor di MiB. Tipe instans EC2 yang berbeda dapat mempengaruhi alokasi memori yang tepat. Statistik simpul yang relevan: Maksimum Statistik klaster yang relevan: Jumlah, Maksimum, Rata-rata
OpenSearchDashboardsHeapUsed	Jumlah absolut memori heap yang digunakan oleh OpenSearch Dasbor di MiB. Statistik simpul yang relevan: Maksimum Statistik klaster yang relevan: Jumlah, Maksimum, Rata-rata
OpenSearchDashboardsHeapUtilization	Persentase maksimum memori heap yang tersedia yang digunakan oleh OpenSearch Dasbor. Jika nilai ini meningkat di atas 80%, pertimbangkan untuk menyesuaikan skala klaster Anda. Statistik simpul yang relevan: Maksimum Statistik klaster yang relevan: Minimum, Maksimum, Rata-rata
OpenSearchDashboardsOS1MinuteLoad	Rata-rata beban CPU satu menit untuk OpenSearch Dasbor. Beban CPU idealnya harus tetap di bawah 1.00. Meskipun lonjakan sementara tidak masalah, Anda sebaiknya meningkatkan ukuran tipe instans jika metrik ini terus di atas 1.00. Statistik simpul yang relevan: Rata-rata Statistik klaster yang relevan: Rata-rata, Maksimum

Metrik	Deskripsi
`OpenSearchDashboardsRequestTotal`	Jumlah total permintaan HTTP yang dibuat ke OpenSearch Dashboards. Jika sistem Anda lambat atau Anda melihat jumlah permintaan Dasbor yang tinggi, pertimbangkan untuk meningkatkan ukuran jenis instans. Statistik simpul yang relevan: Jumlah Statistik klaster yang relevan: Jumlah
`OpenSearchDashboardsResponseTimesMaxInMillis`	Jumlah waktu maksimum, dalam milidetik, yang dibutuhkan OpenSearch Dasbor untuk menanggapi permintaan. Jika permintaan terus memakan waktu lama untuk mengembalikan hasilnya, pertimbangkan untuk meningkatkan ukuran tipe instans. Statistik simpul yang relevan: Maksimum Statistik klaster yang relevan: Maksimum, Rata-rata
`SearchTaskCancelled`	Jumlah pembatalan node koordinator. Statistik simpul yang relevan: Jumlah Statistik klaster yang relevan: Jumlah
`SearchShardTaskCancelled`	Jumlah pembatalan node data. Statistik simpul yang relevan: Jumlah Statistik klaster yang relevan: Jumlah,
`ThreadpoolForce_mergeQueue`	Jumlah antrean tugas dalam kolam utas gabungan daya. Jika ukuran antrean terus tinggi, pertimbangkan untuk menyesuaikan skala klaster Anda. Statistik simpul yang relevan: Maksimum Statistik klaster yang relevan: Jumlah, Maksimum, Rata-rata

Metrik	Deskripsi
ThreadpoolForce_mergeRejected	Jumlah tugas yang ditolak dalam kolam utas gabungan daya. Jika nomor ini terus bertambah, pertimbangkan untuk menyesuaikan skala klaster Anda. Statistik simpul yang relevan: Maksimum Statistik klaster yang relevan: Jumlah
ThreadpoolForce_mergeThreads	Ukuran kolam utas gabungan daya. Statistik simpul yang relevan: Maksimum Statistik klaster yang relevan: Rata-rata, Sum
ThreadpoolIndexQueue	Jumlah antrean tugas dalam kolam utas indeks. Jika ukuran antrean terus tinggi, pertimbangkan untuk menyesuaikan skala klaster Anda. Ukuran antrean indeks maksimum adalah 200. Statistik simpul yang relevan: Maksimum Statistik klaster yang relevan: Jumlah, Maksimum, Rata-rata
ThreadpoolIndexRejected	Jumlah tugas yang ditolak dalam kolam utas indeks. Jika nomor ini terus bertambah, pertimbangkan untuk menyesuaikan skala klaster Anda. Statistik simpul yang relevan: Maksimum Statistik klaster yang relevan: Jumlah
ThreadpoolIndexThreads	Ukuran kolam utas indeks. Statistik simpul yang relevan: Maksimum Statistik klaster yang relevan: Rata-rata, Jumlah

Metrik	Deskripsi
ThreadpoolSearchQueue	Jumlah antrean tugas di kolam utas pencarian. Jika ukuran antrean terus tinggi, pertimbangkan untuk menyesuaikan skala klaster Anda. Ukuran antrean pencarian maksimum adalah 1.000. Statistik simpul yang relevan: Maksimum Statistik klaster yang relevan: Jumlah, Maksimum, Rata-rata
ThreadpoolSearchRejected	Jumlah tugas yang ditolak dalam kolam utas pencarian. Jika nomor ini terus bertambah, pertimbangkan untuk menyesuaikan skala klaster Anda. Statistik simpul yang relevan: Maksimum Statistik klaster yang relevan: Jumlah
ThreadpoolSearchThreads	Ukuran kolam utas pencarian. Statistik simpul yang relevan: Maksimum Statistik klaster yang relevan: Rata-rata, Jumlah
Threadpoolsql-workerQueue	Jumlah antrean tugas di kolam utas pencarian SQL. Jika ukuran antrean terus tinggi, pertimbangkan untuk menyesuaikan skala klaster Anda. Statistik simpul yang relevan: Maksimum Statistik klaster yang relevan: Jumlah, Maksimum, Rata-rata
Threadpoolsql-workerRejected	Jumlah tugas yang ditolak dalam kolam utas pencarian SQL. Jika nomor ini terus bertambah, pertimbangkan untuk menyesuaikan skala klaster Anda. Statistik simpul yang relevan: Maksimum Statistik klaster yang relevan: Jumlah

Metrik	Deskripsi
`Threadpoolsql-workerThreads`	Ukuran kolam utas pencarian SQL. Statistik simpul yang relevan: Maksimum Statistik klaster yang relevan: Rata-rata, Jumlah
`ThreadpoolBulkQueue`	Jumlah antrean tugas dalam kolam utas massal. Jika ukuran antrean terus tinggi, pertimbangkan untuk menyesuaikan skala klaster Anda. Statistik simpul yang relevan: Maksimum Statistik klaster yang relevan: Jumlah, Maksimum, Rata-rata
`ThreadpoolBulkRejected`	Jumlah tugas yang ditolak dalam kolam utas massal. Jika nomor ini terus bertambah, pertimbangkan untuk menyesuaikan skala klaster Anda. Statistik simpul yang relevan: Maksimum Statistik klaster yang relevan: Jumlah
`ThreadpoolBulkThreads`	Ukuran kolam utas massal. Statistik simpul yang relevan: Maksimum Statistik klaster yang relevan: Rata-rata, Jumlah
`ThreadpoolWriteThreads`	Ukuran kolam utas tulis. Statistik simpul yang relevan: Maksimum Statistik klaster yang relevan: Rata-rata, Jumlah
`ThreadpoolWriteQueue`	Jumlah antrean tugas dalam kolam utas tulis. Statistik simpul yang relevan: Maksimum Statistik klaster yang relevan: Rata-rata, Jumlah

Metrik	Deskripsi
`ThreadpoolWriteRejected`	Jumlah tugas yang ditolak dalam kolam utas tulis. Statistik simpul yang relevan: Maksimum Statistik klaster yang relevan: Rata-rata, Sum **Note** Karena ukuran antrian tulis default ditingkatkan dari 200 menjadi 10000 di versi 7.1, metrik ini bukan lagi satu-satunya indikator penolakan dari Layanan. OpenSearch Gunakan `ReplicaWriteRejected` metrik`CoordinatingWriteRejected` ,`PrimaryWriteRejected` , dan untuk memantau penolakan di versi 7.1 dan yang lebih baru.
`CoordinatingWriteRejected`	Jumlah total penolakan terjadi pada node koordinasi karena tekanan pengindeksan sejak startup proses OpenSearch Service terakhir. Statistik simpul yang relevan: Maksimum Statistik klaster yang relevan: Rata-rata, Sum Metrik ini tersedia dalam versi 7.1 ke atas.
`PrimaryWriteRejected`	Jumlah total penolakan terjadi pada pecahan primer karena tekanan pengindeksan sejak startup proses OpenSearch Layanan terakhir. Statistik simpul yang relevan: Maksimum Statistik klaster yang relevan: Rata-rata, Sum Metrik ini tersedia dalam versi 7.1 ke atas.

Metrik	Deskripsi
`ReplicaWriteRejected`	Jumlah total penolakan terjadi pada pecahan replika karena tekanan pengindeksan sejak proses Layanan terakhir OpenSearch dimulai. Statistik simpul yang relevan: Maksimum Statistik klaster yang relevan: Rata-rata, Sum Metrik ini tersedia dalam versi 7.1 ke atas.

UltraWarm metrik

Amazon OpenSearch Service menyediakan metrik berikut untuk UltraWarmnode.

Metrik	Deskripsi
`WarmCPUUtilization`	Persentase penggunaan CPU untuk UltraWarm node di cluster. Maksimum menunjukkan simpul dengan penggunaan CPU tertinggi. Rata-rata mewakili semua UltraWarm node dalam cluster. Metrik ini juga tersedia untuk masing-masing UltraWarm node. Statistik yang relevan: Maksimum, Rata-rata
`WarmFreeStorageSpace`	Jumlah ruang penyimpanan hangat bebas di MiB. Karena UltraWarm menggunakan Amazon S3 daripada disk terlampir, `Sum` adalah satu-satunya statistik yang relevan. Anda harus menyediakan periode pada satu menit untuk mendapatkan nilai yang akurat. Statistik yang relevan: Jumlah
`WarmSearchableDocuments`	Jumlah total dokumen yang dapat dicari di semua indeks hangat di cluster. Anda harus menyediakan periode pada satu menit untuk mendapatkan nilai yang akurat. Statistik yang relevan: Jumlah

Metrik	Deskripsi
`WarmSearchLatency`	Perbedaan total waktu, dalam milidetik, diambil oleh semua pencarian di UltraWarm antara menit N dan menit (N-1). Statistik simpul yang relevan: Rata-rata Statistik klaster yang relevan: Rata-rata, Maksimum
`WarmSearchRate`	Jumlah total permintaan pencarian per menit untuk semua pecahan pada sebuah UltraWarm node. Satu panggilan ke API `_search` mungkin mengembalikan hasilnya dari banyak serpihan yang berbeda.. Jika lima serpihan ini berada pada satu simpul, simpul tersebut akan melaporkan 5 untuk metrik ini, meskipun klien hanya membuat satu permintaan. Statistik simpul yang relevan: Rata-rata Statistik klaster yang relevan: Rata-rata, Maksimum, Jumlah
`WarmStorageSpaceUtilization`	Jumlah total ruang penyimpanan hangat, di MiB, yang digunakan klaster. Statistik yang relevan: Maksimum
`HotStorageSpaceUtilization`	Jumlah total ruang penyimpanan panas yang digunakan klaster. Statistik yang relevan: Maksimum
`WarmSysMemoryUtilization`	Persentase memori simpul hangat yang sedang digunakan. Statistik yang relevan: Maksimum
`HotToWarmMigrationQueueSize`	Jumlah indeks yang saat ini menunggu untuk bermigrasi dari penyimpanan panas ke penyimpanan hangat. Statistik yang relevan: Maksimum
`WarmToHotMigrationQueueSize`	Jumlah indeks yang saat ini menunggu untuk bermigrasi dari penyimpanan hangat ke penyimpanan panas. Statistik yang relevan: Maksimum

Metrik	Deskripsi
`HotToWarm Migration FailureCount`	Jumlah total migrasi panas ke hangat yang gagal. Statistik yang relevan: Jumlah
`HotToWarm Migration ForceMerg eLatency`	Latensi rata-rata tahap gabungan daya dari proses migrasi. Jika tahap ini secara konsisten memakan waktu terlalu lama, pertimbangkan untuk meningkatkan `index.ultrawarm.migration.force_merg e.max_num_segments` . Statistik yang relevan: Rata-rata
`HotToWarm Migration SnapshotL atency`	Latensi rata-rata tahap snapshot dari proses migrasi. Jika tahap ini secara konsisten memakan waktu terlalu lama, pastikan bahwa serpihan Anda tepat ukuran dan didistribusikan di seluruh klaster. Statistik yang relevan: Rata-rata
`HotToWarm Migration Processin gLatency`	Latensi rata-rata pada migrasi panas ke hangat yang sukses, tidak termasuk waktu yang dihabiskan dalam antrean. Nilai ini adalah jumlah dari seluruh waktu yang dibutuhkan untuk menyelesaikan gabungan daya, snapshot, dan tahap relokasi serpihan dari proses migrasi. Statistik yang relevan: Rata-rata
`HotToWarm Migration SuccessCount`	Jumlah total migrasi panas ke hangat yang berhasil. Statistik yang relevan: Jumlah
`HotToWarm Migration SuccessLatency`	Latensi rata-rata pada migrasi panas ke hangat yang sukses, termasuk waktu yang dihabiskan dalam antrean. Statistik yang relevan: Rata-rata
`WarmThrea dpoolSear chThreads`	Ukuran kumpulan utas UltraWarm pencarian. Statistik simpul yang relevan: Maksimum Statistik klaster yang relevan: Rata-rata, Sum

Metrik	Deskripsi
`WarmThreadpoolSearchRejected`	Jumlah tugas yang ditolak di kumpulan thread UltraWarm pencarian. Jika angka ini terus bertambah, pertimbangkan untuk menambahkan lebih banyak UltraWarm node. Statistik simpul yang relevan: Maksimum Statistik klaster yang relevan: Jumlah
`WarmThreadpoolSearchQueue`	Jumlah tugas antrian di kumpulan utas UltraWarm pencarian. Jika ukuran antrian tinggi secara konsisten, pertimbangkan untuk menambahkan lebih banyak UltraWarm node. Statistik simpul yang relevan: Maksimum Statistik klaster yang relevan: Jumlah, Maksimum, Rata-rata
`WarmJVMMemoryPressure`	Persentase maksimum heap Java yang digunakan untuk UltraWarm node. Statistik yang relevan: Maksimum **Note** Logika untuk metrik ini berubah dalam perangkat lunak layanan R20220323. Untuk informasi selengkapnya, lihat catatan rilis.
`WarmOldGenJVMMemoryPressure`	Persentase maksimum heap Java yang digunakan untuk “generasi lama” per UltraWarm node. Statistik yang relevan: Maksimum
`WarmJVMGCYoungCollectionCount`	Berapa kali pengumpulan sampah “generasi muda” berjalan di UltraWarm node. Jumlah eksekusi yang besar dan terus bertambah adalah bagian normal dari operasi klaster. Statistik simpul yang relevan: Maksimum Statistik klaster yang relevan: Jumlah, Maksimum, Rata-rata

Metrik	Deskripsi
`WarmJVMGCYoungCollectionTime`	Jumlah waktu, dalam milidetik, yang dihabiskan cluster untuk melakukan pengumpulan sampah “generasi muda” di UltraWarm node. Statistik simpul yang relevan: Maksimum Statistik klaster yang relevan: Jumlah, Maksimum, Rata-rata
`WarmJVMGCOldCollectionCount`	Berapa kali pengumpulan sampah “generasi lama” berjalan di UltraWarm node. Dalam sebuah klaster dengan sumber daya yang cukup, jumlah ini harus tetap kecil dan jarang bertumbuh. Statistik simpul yang relevan: Maksimum Statistik klaster yang relevan: Jumlah, Maksimum, Rata-rata

Metrik penyimpanan dingin

Amazon OpenSearch Service menyediakan metrik berikut untuk penyimpanan dingin.

Metrik	Deskripsi
`ColdStorageSpaceUtilization`	Jumlah total ruang penyimpanan dingin, di MiB, yang digunakan klaster. Statistik yang relevan: Maks
`ColdToWarmMigrationFailureCount`	Jumlah total migrasi dingin ke hangat yang gagal. Statistik yang relevan: Jumlah
`ColdToWarmMigrationLatency`	Jumlah waktu yang diperlukan untuk berhasil menyelesaikan migrasi dingin ke hangat. Statistik yang relevan: Rata-rata
`ColdToWarmMigrationQueueSize`	Jumlah indeks yang saat ini menunggu untuk bermigrasi dari penyimpanan dingin ke penyimpanan hangat.

Metrik	Deskripsi
	Statistik yang relevan: Maksimum
`ColdToWarmMigrationSuccessCount`	Jumlah total migrasi dingin ke hangat yang berhasil. Statistik yang relevan: Jumlah
`WarmToColdMigrationFailureCount`	Jumlah total migrasi hangat ke dingin yang gagal. Statistik yang relevan: Jumlah
`WarmToColdMigrationLatency`	Jumlah waktu yang diperlukan untuk berhasil menyelesaikan migrasi hangat ke dingin. Statistik yang relevan: Rata-rata
`WarmToColdMigrationQueueSize`	Jumlah indeks yang saat ini menunggu untuk bermigrasi dari penyimpanan hangat ke penyimpanan dingin. Statistik yang relevan: Maksimum
`WarmToColdMigrationSuccessCount`	Jumlah total migrasi hangat ke dingin yang berhasil. Statistik yang relevan: Jumlah

Metrik OR1

Amazon OpenSearch Service menyediakan metrik berikut untuk instans OR1.

Metrik	Deskripsi
`RemoteStorageUsedSpace`	Jumlah total ruang Amazon S3, di MiB, yang digunakan cluster. Statistik yang relevan: Jumlah
`RemoteStorageWriteRejected`	Jumlah total permintaan yang ditolak pada pecahan primer karena penyimpanan jarak jauh dan tekanan replikasi. Ini dihitung mulai dari startup proses OpenSearch Layanan terakhir.

Metrik	Deskripsi
	Statistik yang relevan: Jumlah

Metrik pemberitahuan

Amazon OpenSearch Service menyediakan metrik berikut untuk peringatan.

Metrik	Deskripsi
`AlertingDegraded`	Nilai 1 berarti indeks pemberitahuan berwarna merah atau satu atau beberapa simpul tidak sesuai jadwal. Nilai 0 menunjukkan perilaku normal. Statistik yang relevan: Maksimum
`AlertingIndexExists`	Nilai 1 berarti indeks `.opensearch-alerting-config` tersedia. Nilai 0 berarti indeks tidak tersedia. Sampai Anda menggunakan fitur peringatan untuk pertama kalinya, nilai ini tetap 0. Statistik yang relevan: Maksimum
`AlertingIndexStatus.green`	Kesehatan indeks. Nilai 1 berarti hijau. Nilai 0 berarti indeks tidak tersedia atau tidak hijau. Statistik yang relevan: Maksimum
`AlertingIndexStatus.red`	Kesehatan indeks. Nilai 1 berarti merah. Nilai 0 berarti indeks tidak tersedia atau tidak merah. Statistik yang relevan: Maksimum
`AlertingIndexStatus.yellow`	Kesehatan indeks. Nilai 1 berarti kuning. Nilai 0 berarti indeks tidak tersedia atau tidak kuning. Statistik yang relevan: Maksimum
`AlertingNodesNotOnSchedule`	Nilai 1 berarti beberapa tugas tidak berjalan sesuai jadwal. Nilai 0 berarti semua tugas pemberitahuan berjalan sesuai jadwal (atau bahwa tidak terdapat tugas pemberitahuan). Periksa konsol OpenSearch Layanan

Metrik	Deskripsi
	atau buat `_nodes/stats` permintaan untuk melihat apakah ada node yang menunjukkan penggunaan sumber daya yang tinggi. Statistik yang relevan: Maksimum
`AlertingNodesOnSchedule`	Nilai 1 berarti semua tugas pemberitahuan berjalan sesuai jadwal (atau bahwa tidak terdapat tugas pemberitahuan). Nilai 0 berarti beberapa tugas tidak berjalan sesuai jadwal. Statistik yang relevan: Maksimum
`AlertingScheduledJobEnabled`	Nilai 1 berarti Pengaturan klaster `opensearch.scheduled_jobs.enabled` betul. Nilai 0 berarti itu adalah salah, dan tugas yang dijadwalkan dinonaktifkan. Statistik yang relevan: Maksimum

Metrik deteksi anomali

Amazon OpenSearch Service menyediakan metrik berikut untuk deteksi anomali.

Metrik	Deskripsi
`ADPluginUnhealthy`	Nilai 1 berarti bahwa plugin deteksi anomali tidak berfungsi dengan baik, baik karena jumlah kegagalan yang tinggi atau karena salah satu indeks yang digunakannya berwarna merah. Nilai 0 menunjukkan plugin bekerja seperti yang diharapkan. Statistik yang relevan: Maksimum
`ADExecuteRequestCount`	Jumlah permintaan untuk mendeteksi anomali. Statistik yang relevan: Jumlah
`ADExecuteFailureCount`	Jumlah permintaan gagal untuk mendeteksi anomali. Statistik yang relevan: Jumlah

Metrik	Deskripsi
ADHCExecuteFailureCount	Jumlah permintaan gagal untuk mendeteksi anomali pada detektor kardinalitas tinggi. Statistik yang relevan: Jumlah
ADHCExecuteRequestCount	Jumlah permintaan untuk mendeteksi anomali pada detektor kardinalitas tinggi. Statistik yang relevan: Jumlah
ADAnomalyResultsIndexStatusIndexExists	Nilai 1 berarti indeks yang menurut alias `.opensearch-anomaly-results` tersedia. Sampai Anda menggunakan deteksi anomali untuk pertama kalinya, nilai ini tetap 0. Statistik yang relevan: Maksimum
ADAnomalyResultsIndexStatus.red	Nilai 1 berarti indeks yang menurut alias `.opensearch-anomaly-results` berwarna merah. Nilai 0 berarti indeks tidak berwarna merah. Sampai Anda menggunakan deteksi anomali untuk pertama kalinya, nilai ini tetap 0. Statistik yang relevan: Maksimum
ADAnomalyDetectorsIndexStatusIndexExists	Nilai 1 berarti indeks `.opensearch-anomaly-detectors` tersedia. Nilai 0 berarti indeks tidak tersedia. Sampai Anda menggunakan deteksi anomali untuk pertama kalinya, nilai ini tetap 0. Statistik yang relevan: Maksimum
ADAnomalyDetectorsIndexStatus.red	Nilai 1 berarti indeks `.opensearch-anomaly-detectors` berwarna merah. Nilai 0 berarti indeks tidak berwarna merah. Sampai Anda menggunakan deteksi anomali untuk pertama kalinya, nilai ini tetap 0. Statistik yang relevan: Maksimum

Metrik	Deskripsi
`ADModelsCheckpointIndexStatusIndexExists`	Nilai 1 berarti indeks `.opensearch-anomaly-checkpoints` tersedia. Nilai 0 berarti indeks tidak tersedia. Sampai Anda menggunakan deteksi anomali untuk pertama kalinya, nilai ini tetap 0. Statistik yang relevan: Maksimum
`ADModelsCheckpointIndexStatus.red`	Nilai 1 berarti indeks `.opensearch-anomaly-checkpoints` berwarna merah. Nilai 0 berarti indeks tidak berwarna merah. Sampai Anda menggunakan deteksi anomali untuk pertama kalinya, nilai ini tetap 0. Statistik yang relevan: Maksimum

Metrik pencarian asinkron

Amazon OpenSearch Service menyediakan metrik berikut untuk pencarian asinkron.

Statistik simpul koordinator pencarian asinkron (per node koordinator)

Metrik	Deskripsi
`AsynchronousSearchSubmissionRate`	Jumlah pencarian asinkron yang dikirimkan di menit terakhir.
`AsynchronousSearchInitializedRate`	Jumlah pencarian asinkron yang diinisialisasi di menit terakhir.
`AsynchronousSearchRunningCurrent`	Jumlah pencarian asinkron yang saat ini berjalan.

Metrik	Deskripsi
AsynchronousSearchCompletionRate	Jumlah pencarian asinkron yang berhasil diselesaikan di menit terakhir.
AsynchronousSearchFailureRate	Jumlah pencarian asinkron yang diselesaikan dan gagal di menit terakhir.
AsynchronousSearchPersistRate	Jumlah pencarian asinkron yang bertahan di menit terakhir.
AsynchronousSearchPersistFailedRate	Jumlah pencarian asinkron yang tidak bertahan di menit terakhir.
AsynchronousSearchRejected	Jumlah total pencarian asinkron yang ditolak sejak waktu aktif simpul.
AsynchronousSearchCancelled	Jumlah total pencarian asinkron yang dibatalkan sejak waktu aktif simpul.
AsynchronousSearchMaxRunningTime	Durasi pencarian asinkron terpanjang berjalan pada simpul di menit terakhir.

Statistik klaster pencarian asinkron

Metrik	Deskripsi
`AsynchronousSearchStoreHealth`	Kondisi penyimpanan dalam indeks bertahan (MERAH/non-MERAH) di menit terakhir.
`AsynchronousSearchStoreSize`	Ukuran indeks sistem pada semua serpihan di menit terakhir.
`AsynchronousSearchStoredResponseCount`	Jumlah tanggapan yang tersimpan dalam indeks sistem di menit terakhir.

Metrik Penyetelan Otomatis

Amazon OpenSearch Service menyediakan metrik berikut untuk Auto-Tune.

Metrik	Deskripsi
`AutoTuneChangesHistoryHeapSize`	Riwayat perubahan di MiB untuk nilai penyetelan ukuran heap.
`AutoTuneChangesHistoryJVMYoungGenArgs`	Riwayat perubahan untuk argumen JVM YongGen .
`AutoTuneFailed`	Boolean yang menunjukkan jika perubahan Auto-Tune gagal.
`AutoTuneSucceeded`	Boolean yang menunjukkan apakah perubahan Auto-Tune berhasil.
`AutoTuneValue`	Riwayat perubahan antrian (hitungan) dan riwayat perubahan penyetelan cache (di MiB) untuk perubahan yang tidak mengganggu.

Multi-AZ dengan metrik Siaga

Amazon OpenSearch Service menyediakan metrik berikut untuk Multi-AZ dengan Standby.

Metrik tingkat simpul untuk node data di Availability Zone aktif

Metrik	Deskripsi
`CPUUtilization`	Persentase penggunaan CPU untuk simpul data di klaster. Maksimum menunjukkan simpul dengan penggunaan CPU tertinggi. Rata-rata mewakili semua simpul dalam klaster. Metrik ini juga tersedia untuk masing-masing simpul.
`FreeStorageSpace`	Ruang kosong untuk simpul data dalam klaster. `Sum` menunjukkan total ruang kosong untuk klaster, tetapi Anda harus meninggalkan periode pada satu menit untuk mendapatkan nilai yang akurat. `Minimum` dan `Maximum` menunjukkan simpul dengan ruang paling sedikit dan paling bebas, menurut urutannya. Metrik ini juga tersedia untuk masing-masing node. OpenSearch Layanan melempar a `ClusterBlockException` saat metrik ini mencapai`0`. Untuk memulihkan, Anda harus menghapus indeks, menambahkan instance yang lebih besar, atau menambahkan penyimpanan berbasis EBS ke instance yang ada. Untuk mempelajari selengkapnya, lihat the section called “Kurangnya ruang penyimpanan yang tersedia”. Konsol OpenSearch Layanan menampilkan nilai ini di GiB. CloudWatch Konsol Amazon menampilkannya di MiB.
`JVMMemoryPressure`	Persentase maksimum heap Java yang digunakan untuk semua node data di cluster. OpenSearch Layanan menggunakan setengah dari RAM instance untuk heap Java, hingga ukuran heap 32 GiB. Anda dapat menskalakan instans secara vertikal hingga 64 GiB RAM, di mana Anda dapat menskalakan secara horizontal dengan menambahkan instans. Lihat the section called “ CloudWatch Alarm yang direkomendasikan”.
`SysMemoryUtilization`	Persentase memori instans yang sedang digunakan. Nilai tinggi untuk metrik ini normal dan biasanya tidak mewakili masalah dengan klaster

Metrik	Deskripsi
	Anda. Untuk indikator yang lebih baik mengenai potensi masalah performa dan stabilitas, lihat metrik `JVMMemoryPressure` .
`IndexingLatency`	Perbedaan total waktu, dalam milidetik, diambil oleh semua operasi pengindeksan dalam simpul antara menit N dan menit (N-1).
`IndexingRate`	Jumlah operasi pengindeksan per menit.
`SearchLatency`	Perbedaan total waktu, dalam milidetik, diambil oleh semua pencarian dalam simpul antara menit N dan menit (N-1).
`SearchRate`	Jumlah total permintaan pencarian per menit untuk semua serpihan pada simpul data.
`ThreadpoolSearchQueue`	Jumlah antrean tugas di kolam utas pencarian. Jika ukuran antrean terus tinggi, pertimbangkan untuk menyesuaikan skala klaster Anda. Ukuran antrean pencarian maksimum adalah 1.000.
`ThreadpoolWriteQueue`	Jumlah antrean tugas dalam kolam utas tulis.
`ThreadpoolSearchRejected`	Jumlah tugas yang ditolak dalam kolam utas pencarian. Jika nomor ini terus bertambah, pertimbangkan untuk menyesuaikan skala klaster Anda.
`ThreadpoolWriteRejected`	Jumlah tugas yang ditolak dalam kolam utas tulis.

Metrik tingkat cluster untuk cluster di Availability Zone aktif

Metrik	Deskripsi
`DataNodes`	Jumlah total pecahan aktif dan siaga.
`DataNodesShards.active`	Jumlah total aktif serpihan primer dan replika aktif.

Metrik	Deskripsi
`DataNodes Shards.unassigned`	Jumlah serpihan yang tidak dialokasikan ke simpul di klaster.
`DataNodes Shards.initializing`	Jumlah serpihan yang berada di bawah inisialisasi.
`DataNodes Shards.relocating`	Jumlah serpihan yang berada di bawah relokasi.

Metrik rotasi Zona Ketersediaan

JikaActiveReads.*Availability-Zone* = 1, maka zona tersebut aktif.
JikaActiveReads.*Availability-Zone* = 0, maka zona dalam keadaan siaga.

Metrik titik dalam waktu

Amazon OpenSearch Service menyediakan metrik berikut untuk pencarian point in time (PIT).

Statistik simpul koordinator PIT (per node koordinator)

Metrik	Deskripsi
`CurrentPointInTime`	Jumlah konteks pencarian PIT aktif di node.
`TotalPointInTime`	Jumlah konteks pencarian PIT yang kedaluwarsa sejak node up time.
`AvgPointInTimeAliveTime`	Rata-rata tetap hidup dari konteks pencarian PIT sejak node up time.
`HasActivePointInTime`	Nilai 1 menunjukkan bahwa ada konteks PIT aktif pada node sejak waktu naik node. Nilai 0 berarti tidak ada.

Metrik	Deskripsi
`HasUsedPointInTime`	Nilai 1 menunjukkan bahwa ada konteks PIT kedaluwarsa pada node sejak waktu habis node. Nilai 0 berarti tidak ada.

Metrik SQL

Amazon OpenSearch Service menyediakan metrik berikut untuk dukungan SQL.

Metrik	Deskripsi
`SQLFailedRequestCountByCusErr`	Jumlah permintaan untuk API `_sql` yang gagal karena masalah klien. Sebagai contoh, permintaan mungkin mengembalikan kode status HTTP 400 karena `IndexNotFoundException` . Statistik yang relevan: Jumlah
`SQLFailedRequestCountBySysErr`	Jumlah permintaan untuk API `_sql` yang gagal karena masalah server atau pembatasan fitur. Sebagai contoh, permintaan mungkin mengembalikan kode status HTTP 503 karena `VerificationException` . Statistik yang relevan: Jumlah
`SQLRequestCount`	Jumlah permintaan untuk API `_sql`. Statistik yang relevan: Jumlah
`SQLDefaultCursorRequestCount`	Mirip dengan`SQLRequestCount` , tetapi hanya menghitung permintaan pagination. Statistik yang relevan: Jumlah
`SQLUnhealthy`	Nilai 1 menunjukkan bahwa, dalam menanggapi permintaan tertentu, plugin SQL mengembalikan kode respons 5 xx atau meneruskan kueri DSL yang tidak valid ke. OpenSearch Permintaan lainnya harus terus berhasil. Nilai 0 menunjukkan tidak ada kegagalan baru-baru ini. Jika Anda melihat nilai berkelanjutan 1, pecahkan masalah permintaan yang klien Anda buat ke plugin.

Metrik	Deskripsi
	Statistik yang relevan: Maksimum

metrik k-NN

Amazon OpenSearch Service menyertakan metrik berikut untuk plugin k-nearest neighbor (k-NN).

Metrik	Deskripsi
`KNNCacheCapacityReached`	Metrik per simpul untuk melihat apakah kapasitas cache telah tercapai. Metrik ini hanya relevan dengan perkiraan pencarian k-NN. Statistik yang relevan: Maksimum
`KNNCircuitBreakerTriggered`	Metrik per-klaster untuk melihat apakah pemutus sirkuit dipicu. Jika terdapat simpul yang mengembalikan nilai 1 untuk `KNNCacheCapacityReached` , nilai ini juga akan mengembalikan 1. Metrik ini hanya relevan dengan perkiraan pencarian k-NN. Statistik yang relevan: Maksimum
`KNNEvictionCount`	Metrik per-simpul untuk sejumlah grafik yang telah dikosongkan dari cache karena kendala memori atau waktu siaga. Pengosongan eksplisit yang terjadi karena penghapusan indeks tidak diperhitungkan. Metrik ini hanya relevan dengan perkiraan pencarian k-NN. Statistik yang relevan: Jumlah
`KNNGraphIndexErrors`	Metrik per simpul pada sejumlah permintaan untuk menambahkan `knn_vector` bidang dokumen ke grafik yang menghasilkan kesalahan. Statistik yang relevan: Jumlah

Metrik	Deskripsi
KNNGraphIndexRequests	Metrik per simpul pada sejumlah permintaan untuk menambahk an knn_vector bidang dokumen ke grafik. Statistik yang relevan: Jumlah
KNNGraphMemoryUsage	Metrik per simpul pada ukuran cache saat ini (ukuran total semua grafik dalam memori) dalam kilobyte. Metrik ini hanya relevan dengan perkiraan pencarian k-NN. Statistik yang relevan: Rata-rata
KNNGraphQueryErrors	Metrik per simpul untuk sejumlah kueri grafik yang menghasil kan kesalahan. Statistik yang relevan: Jumlah
KNNGraphQueryRequests	Metrik per simpul untuk sejumlah kueri grafik. Statistik yang relevan: Jumlah
KNNHitCount	Metrik per simpul untuk sejumlah temuan cache. Sebuah temuan cache terjadi ketika pengguna mengajukan kueri grafik yang sudah dimuat ke dalam memori. Metrik ini hanya relevan dengan perkiraan pencarian k-NN. Statistik yang relevan: Jumlah
KNNLoadExceptionCount	Metrik per simpul untuk beberapa kali pengecualian timbul ketika mencoba untuk memuat grafik ke dalam cache. Metrik ini hanya relevan dengan perkiraan pencarian k-NN. Statistik yang relevan: Jumlah
KNNLoadSuccessCount	Metrik per simpul untuk frekuensi ketika plugin berhasil memuat grafik ke dalam cache. Metrik ini hanya relevan dengan perkiraan pencarian k-NN. Statistik yang relevan: Jumlah

Metrik	Deskripsi
KNNMissCount	Metrik per simpul untuk sejumlah kelalaian cache. Kelalaian cache terjadi ketika pengguna mengajukan kueri grafik yang belum dimuat ke dalam memori. Metrik ini hanya relevan dengan perkiraan pencarian k-NN. Statistik yang relevan: Jumlah
KNNQueryRequests	Metrik per simpul untuk sejumlah permintaan kueri yang diterima plugin k-NN. Statistik yang relevan: Jumlah
KNNScriptCompilationErrors	Metrik per simpul untuk sejumlah kesalahan selama kompilasi penulisan. Statistik ini hanya relevan dengan pencarian penulisan skor k-NN. Statistik yang relevan: Jumlah
KNNScriptCompilations	Metrik per simpul untuk frekuensi kompilasi penulisan k-NN. Nilai ini biasanya harus 1 atau 0, tetapi jika cache yang berisi kompilasi penulisan telah terisi, penulisan k-NN dapat dikompilasi ulang. Statistik ini hanya relevan dengan pencarian penulisan skor k-NN. Statistik yang relevan: Jumlah
KNNScriptQueryErrors	Metrik per simpul untuk sejumlah kesalahan selama kueri penulisan. Statistik ini hanya relevan dengan pencarian penulisan skor k-NN. Statistik yang relevan: Jumlah
KNNScriptQueryRequests	Metrik per simpul untuk sejumlah total kueri penulisan. Statistik ini hanya relevan dengan pencarian penulisan skor k-NN. Statistik yang relevan: Jumlah

Metrik	Deskripsi
KNNTotalLoadTime	Waktu dalam nanodetik yang diperlukan k-NN untuk memuat grafik ke dalam cache. Metrik ini hanya relevan dengan perkiraan pencarian k-NN. Statistik yang relevan: Jumlah

Metrik pencarian lintas klaster

Amazon OpenSearch Service menyediakan metrik berikut untuk pencarian lintas klaster.

Metrik domain sumber

Metrik	Dimensi	Deskripsi
CrossClusterOutboundConnections	ConnectionId	Jumlah simpul yang terhubung. Jika respons Anda mencakup satu atau beberapa domain yang dilewati, gunakan metrik ini untuk melacak koneksi yang tidak sehat. Jika nomor ini turun menjadi 0, maka koneksi tidak sehat.
CrossClusterOutboundRequests	ConnectionId	Jumlah permintaan pencarian yang dikirim ke domain tujuan. Gunakan ini untuk memeriksa apakah beban permintaan pencarian lintas klaster membanjiri domain Anda, korelasikan lonjakan apa pun dalam metrik ini dengan lonjakan JVM/CPU.

Metrik domain tujuan

Metrik	Dimensi	Deskripsi
CrossClusterInboundRequests	ConnectionId	Jumlah permintaan koneksi masuk yang diterima dari domain sumber.

Tambahkan CloudWatch alarm jika Anda kehilangan koneksi secara tidak terduga. Untuk langkah-langkah membuat alarm, lihat Membuat CloudWatch Alarm Berdasarkan Ambang Statis.

Metrik replikasi lintas-cluster

Amazon OpenSearch Service menyediakan metrik berikut untuk replikasi lintas cluster.

Metrik	Deskripsi
`ReplicationRate`	Tingkat rata-rata operasi replikasi per detik. Metrik ini mirip dengan `IndexingRate` metrik.
`LeaderCheckPoint`	Untuk koneksi tertentu, jumlah nilai pos pemeriksaan pemimpin di semua indeks yang mereplikasi. Anda dapat menggunakan metrik ini untuk mengukur latensi replikasi.
`FollowerCheckPoint`	Untuk koneksi tertentu, jumlah nilai pos pemeriksaan pengikut di semua indeks yang mereplikasi. Anda dapat menggunakan metrik ini untuk mengukur latensi replikasi.
`ReplicationNumSyncingIndices`	Jumlah indeks yang memiliki status replikasi. `SYNCING`
`ReplicationNumBootstrappingIndices`	Jumlah indeks yang memiliki status replikasi. `BOOTSTRAPPING`
`ReplicationNumPausedIndices`	Jumlah indeks yang memiliki status replikasi. `PAUSED`
`ReplicationNumFailedIndices`	Jumlah indeks yang memiliki status replikasi. `FAILED`
`CrossClusterOutbou`	Jumlah permintaan transpor replikasi pada domain pengikut. Permintaan transportasi bersifat internal dan terjadi setiap kali operasi API

Metrik	Deskripsi
ndReplica tionRequests	replikasi dipanggil. Mereka juga terjadi ketika polling domain pengikut berubah dari domain pemimpin.
CrossClus terInboun dReplicat ionRequests	Jumlah permintaan transpor replikasi pada domain pemimpin. Permintaan transportasi bersifat internal dan terjadi setiap kali operasi API replikasi dipanggil.
AutoFollo wNumSucce ssStartRe plication	Jumlah indeks pengikut yang telah berhasil dibuat oleh aturan replikasi untuk koneksi tertentu.
AutoFollo wNumFaile dStartRep lication	Jumlah indeks pengikut yang gagal dibuat oleh aturan replikasi ketika ada pola yang cocok. Masalah ini mungkin timbul karena masalah jaringan di cluster jarak jauh, atau masalah keamanan (yaitu peran terkait tidak memiliki izin untuk memulai replikasi).
AutoFollo wLeaderCa llFailure	Apakah ada kueri yang gagal dari indeks pengikut ke indeks pemimpin untuk menarik data baru. Nilai 1 berarti bahwa ada 1 atau lebih panggilan gagal di menit terakhir.

Metrik Learning to Rank

Amazon OpenSearch Service menyediakan metrik berikut untuk Belajar Peringkat.

Metrik	Deskripsi
LTRReques tTotalCount	Jumlah total permintaan peringkat.
LTRReques tErrorCount	Jumlah total permintaan gagal.
LTRStatus.red	Melacak jika salah satu indeks yang diperlukan untuk menjalankan plugin berwarna merah.

Metrik	Deskripsi
LTRMemoryUsage	Total memori yang digunakan oleh plugin.
LTRFeaturMemoryUsageInBytes	Jumlah memori, dalam byte, yang digunakan oleh bidang fitur Learning to Rank.
LTRFeaturesetMemoryUsageInBytes	Jumlah memori, dalam byte, yang digunakan oleh seluruh set fitur Learning to Rank.
LTRModelMemoryUsageInBytes	Jumlah memori, dalam byte, yang digunakan oleh seluruh model Learning to Rank.

Metrik Bahasa Pemrosesan yang Disalurkan

Amazon OpenSearch Service menyediakan metrik berikut untuk Bahasa Pemrosesan Piped.

Metrik	Deskripsi
PPLFailedRequestCountByCusErr	Jumlah permintaan untuk API _ppl yang gagal karena masalah klien. Sebagai contoh, permintaan mungkin mengembalikan kode status HTTP 400 karena IndexNotFoundException .
PPLFailedRequestCountBySysErr	Jumlah permintaan untuk API _ppl yang gagal karena masalah server atau pembatasan fitur. Sebagai contoh, permintaan mungkin mengembalikan kode status HTTP 503 karena VerificationException .
PPLRequestCount	Jumlah permintaan untuk API _ppl.

Memantau OpenSearch log dengan Amazon CloudWatch Logs

Amazon OpenSearch Service mengekspos OpenSearch log berikut melalui Amazon CloudWatch Logs:

- Log kesalahan
- Log lambat
- Log audit

Pencarian log lambat, indeks log lambat, dan log kesalahan berguna untuk memecahkan masalah performa dan stabilitas. Log audit melacak aktivitas pengguna untuk tujuan kepatuhan. Semua log bersifat nonaktif secara default. Jika diaktifkan, CloudWatchharga standar berlaku.

> **Note**
>
> Log kesalahan tersedia hanya untuk OpenSearch dan Elasticsearch versi 5.1 dan lebih baru. Log lambat tersedia untuk semua OpenSearch dan versi Elasticsearch.

Untuk log-nya, OpenSearch menggunakan Apache Log4j 2 dan tingkat log bawaan (dari yang teringan sampai yang terparah) dari`TRACE`,,,,,,,`DEBUG`, `INFO``WARN`, `ERROR` dan. `FATAL`

Jika Anda mengaktifkan log kesalahan, OpenSearch Layanan menerbitkan baris log dari`WARN`,`ERROR`, dan `FATAL` untukCloudWatch. OpenSearchLayanan juga menerbitkan beberapa pengecualian dari `DEBUG` tingkat, termasuk berikut ini:

- `org.opensearch.index.mapper.MapperParsingException`
- `org.opensearch.index.query.QueryShardException`
- `org.opensearch.action.search.SearchPhaseExecutionException`
- `org.opensearch.common.util.concurrent.OpenSearchRejectedExecutionException`
- `java.lang.IllegalArgumentException`

Log kesalahan dapat membantu pemecahan masalah dalam banyak situasi, termasuk berikut ini:

- Masalah kompilasi penulisan tanpa rasa sakit
- Kueri tidak valid
- Masalah pengindeksan
- Kegagalan snapshot
- Manajemen State State State State State State State State State State State State Indeks

Topik

Mengaktifkan penerbitan log (konsol)

Konsol OpenSearch Layanan adalah cara termudah untuk mengaktifkan penerbitan log. CloudWatch

Untuk mengaktifkan penerbitan log ke CloudWatch (konsol)

1. Masuk ke https://aws.amazon.com, dan kemudian pilih Masuk ke Konsol.
2. Di bawah Analytics, pilih OpenSearchLayanan Amazon.
3. Pilih domain yang ingin Anda perbarui.
4. Pada tab Log, pilih jenis log dan pilih Aktifkan.
5. Buat grup CloudWatch log baru atau pilih grup yang sudah ada.

> **Note**
>
> Jika Anda berencana untuk mengaktifkan beberapa log, sebaiknya penerbitan masing-masing dilakukan ke grup log sendiri. Pemisahan ini membuat log lebih mudah untuk dipindai.

6. Pilih kebijakan akses yang berisi izin yang sesuai, atau buat kebijakan dengan menggunakan JSON yang disediakan oleh konsol:

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

```
      "Action": [
        "logs:PutLogEvents",
        "logs:CreateLogStream"
      ],
      "Resource": "cw_log_group_arn:*"
    }
  ]
}
```

Kami menyarankan Anda menambahkan `aws:SourceAccount` dan kunci `aws:SourceArn` kondisi ke kebijakan untuk melindungi diri Anda dari masalah wakil yang bingung. Akun sumber adalah pemilik domain dan sumber ARN adalah ARN domain. Domain Anda harus menggunakan perangkat lunak layanan R20211203 atau yang lebih baru untuk menambahkan kunci kondisi ini.

Misalnya, Anda dapat menambahkan blok kondisi berikut ke kebijakan:

```
"Condition": {
    "StringEquals": {
        "aws:SourceAccount": "account-id"
    },
    "ArnLike": {
        "aws:SourceArn": "arn:aws:es:region:account-id:domain/domain-name"
    }
}
```

> **Important**
>
> CloudWatchDukungan log 10 kebijakan sumber daya per Wilayah. Jika Anda berencana untuk mengaktifkan log untuk beberapa domain OpenSearch Layanan, Anda harus membuat dan menggunakan kembali kebijakan yang lebih luas yang mencakup beberapa grup log agar tidak mencapai batas ini. Untuk langkah-langkah memperbarui kebijakan, lihat the section called “Mengaktifkan penerbitan log (AWS CLI)”.

7. Pilih Aktifkan.

Status domain Anda berubah dari Aktif ke Pemrosesan. Status harus kembali ke Aktif sebelum penerbitan log diaktifkan. Perubahan ini umumnya memakan waktu 30 menit, tetapi dapat lebih lama tergantung pada konfigurasi domain Anda.

Jika Anda mengaktifkan salah satu log lambat, lihat the section called “Pengaturan ambang OpenSearch logging untuk log lambat”. Jika Anda mengaktifkan log audit, lihat the section called “Langkah 2: Aktifkan log audit di OpenSearch Dasbor”. Jika Anda mengaktifkan log kesalahan saja, Anda tidak perlu melakukan langkah konfigurasi tambahan apa pun.

Mengaktifkan penerbitan log (AWS CLI)

Agar dapat mengaktifkan penerbitan log, Anda memerlukan grup CloudWatch log. Jika Anda belum memilikinya, Anda dapat membuatnya dengan menggunakan perintah berikut:

```
aws logs create-log-group --log-group-name my-log-group
```

Masukkan perintah berikutnya untuk menemukan ARN grup log, lalu buat catatan tentang hal tersebut:

```
aws logs describe-log-groups --log-group-name my-log-group
```

Sekarang Anda dapat memberikan izin OpenSearch Service untuk menulis ke grup log. Anda harus memberikan ARN grup log dekat akhir perintah:

```
aws logs put-resource-policy \
  --policy-name my-policy \
  --policy-document '{ "Version": "2012-10-17", "Statement": [{ "Sid": "",
 "Effect": "Allow", "Principal": { "Service": "es.amazonaws.com"}, "Action":
[ "logs:PutLogEvents","logs:CreateLogStream"],"Resource": "cw_log_group_arn:*"}]}'
```

> **Important**
>
> CloudWatchDukungan log 10 kebijakan sumber daya per Wilayah. Jika Anda berencana untuk mengaktifkan log lambat untuk beberapa domain OpenSearch Layanan, Anda harus membuat dan menggunakan kembali kebijakan yang lebih luas yang mencakup beberapa grup log agar tidak mencapai batas ini.

Jika Anda perlu meninjau kebijakan ini di lain waktu, gunakan Perintah `aws logs describe-resource-policies`. Untuk memperbarui kebijakan, terbitkan perintah `aws logs put-resource-policy` yang sama dengan dokumen kebijakan baru.

Akhirnya, Anda dapat menggunakan opsi `--log-publishing-options` untuk mengaktifkan penerbitan. Sintaks untuk opsi tersebut sama-sama digunakan untuk perintah `create-domain` dan `update-domain-config`.

Parameter	Nilai valid
`--log-publishing-options`	`SEARCH_SLOW_LOGS={CloudWatchLogsLogGroupArn= cw_log_group_arn ,Enabled=true|false}`
	`INDEX_SLOW_LOGS={CloudWatchLogsLogGroupArn= cw_log_group_arn ,Enabled=true|false}`
	`ES_APPLICATION_LOGS={CloudWatchLogsLogGroupArn= cw_log_group_arn ,Enabled=true|false}`
	`AUDIT_LOGS={CloudWatchLogsLogGroupArn= cw_log_group_arn ,Enabled=true|false}`

> **Note**
>
> Jika Anda berencana untuk mengaktifkan beberapa log, sebaiknya penerbitan masing-masing dilakukan ke grup log sendiri. Pemisahan ini membuat log lebih mudah untuk dipindai.

Contoh

Contoh berikut memungkinkan penerbitan pencarian dan pengindeks log lambat untuk domain tertentu:

```
aws opensearch update-domain-config \
  --domain-name my-domain \
  --log-publishing-options
 "SEARCH_SLOW_LOGS={CloudWatchLogsLogGroupArn=arn:aws:logs:us-east-1:123456789012:log-group:my-log-
group,Enabled=true},INDEX_SLOW_LOGS={CloudWatchLogsLogGroupArn=arn:aws:logs:us-east-1:123456789012:log-group:my-other-log-group,Enabled=true}"
```

Untuk menonaktifkan penerbitanCloudWatch, jalankan perintah yang sama dengan`Enabled=false`.

Jika Anda mengaktifkan salah satu log lambat, lihat the section called “Pengaturan ambang OpenSearch logging untuk log lambat”. Jika Anda mengaktifkan log audit, lihat the section called “Langkah 2: Aktifkan log audit di OpenSearch Dasbor”. Jika Anda mengaktifkan log kesalahan saja, Anda tidak perlu melakukan langkah konfigurasi tambahan apa pun.

Mengaktifkan penerbitan log (AWS SDK)

Agar dapat mengaktifkan penerbitan log, Anda harus terlebih dahulu membuat grup CloudWatch log, mendapatkan ARN-nya, lalu memberikan izin OpenSearch Service untuk menuliskannya. Operasi yang relevan didokumentasikan dalam Referensi API Amazon CloudWatch Logs:

- `CreateLogGroup`
- `DescribeLogGroup`
- `PutResourcePolicy`

Anda dapat mengakses operasi ini dengan menggunakan AWS SDK.

AWSSDK (kecuali SDK Android dan iOS) mendukung semua operasi yang ditetapkan dalam Referensi API Amazon OpenSearch Service, termasuk `--log-publishing-options` opsi untuk `CreateDomain` dan. `UpdateDomainConfig`

Jika Anda mengaktifkan salah satu log lambat, lihat the section called “Pengaturan ambang OpenSearch logging untuk log lambat”. Jika Anda mengaktifkan log kesalahan saja, Anda tidak perlu melakukan langkah konfigurasi tambahan apa pun.

Mengaktifkan penerbitan log (CloudFormation)

Dalam contoh ini, kami menggunakan CloudFormation untuk membuat grup log yang disebut`opensearch-logs`, menetapkan izin yang sesuai, lalu membuat domain dengan penerbitan log yang diaktifkan untuk log aplikasi, pencarian log lambat, dan log lambat indeks.

Agar dapat mengaktifkan penerbitan log, Anda perlu membuat grup CloudWatch log:

```
Resources:
  OpenSearchLogGroup:
    Type: AWS::Logs::LogGroup
    Properties:
      LogGroupName: opensearch-logs
Outputs:
  Arn:
```

```
    Value:
      'Fn::GetAtt':
        - OpenSearchLogGroup
        - Arn
```

Template menghasilkan ARN grup log. Dalam hal ini, ARN adalah`arn:aws:logs:us-east-1:123456789012:log-group:opensearch-logs`.

Menggunakan ARN, buat kebijakan sumber daya yang memberikan izin OpenSearch Service untuk menulis ke grup log:

```
Resources:
 OpenSearchLogPolicy:
   Type: AWS::Logs::ResourcePolicy
   Properties:
     PolicyName: my-policy
     PolicyDocument: "{ \"Version\": \"2012-10-17\", \"Statement\": [{ \"Sid\": \"\",
 \"Effect\": \"Allow\", \"Principal\": { \"Service\": \"es.amazonaws.com\"}, \"Action
\":[ \"logs:PutLogEvents\",\"logs:CreateLogStream\"],\"Resource\": \"arn:aws:logs:us-
east-1:123456789012:log-group:opensearch-logs:*\"}]}"
```

Terakhir, buat CloudFormation tumpukan berikut, yang menghasilkan domain OpenSearch Service dengan penerbitan log. Kebijakan akses mengizinkan pengguna untuk membuat semua permintaan HTTP ke domain. Akun AWS

```
Resources:
  OpenSearchServiceDomain:
    Type: "AWS::OpenSearchService::Domain"
    Properties:
      DomainName: my-domain
      EngineVersion: "OpenSearch_1.0"
      ClusterConfig:
        InstanceCount: 2
        InstanceType: "r6g.xlarge.search"
        DedicatedMasterEnabled: true
        DedicatedMasterCount: 3
        DedicatedMasterType: "r6g.xlarge.search"
      EBSOptions:
        EBSEnabled: true
        VolumeSize: 10
        VolumeType: "gp2"
      AccessPolicies:
```

```
        Version: "2012-10-17"
        Statement:
            Effect: "Allow"
            Principal:
                AWS: "arn:aws:iam::123456789012:user/es-user"
            Action: "es:*"
            Resource: "arn:aws:es:us-east-1:123456789012:domain/my-domain/*"
      LogPublishingOptions:
        ES_APPLICATION_LOGS:
          CloudWatchLogsLogGroupArn: "arn:aws:logs:us-east-1:123456789012:log-
group:opensearch-logs"
          Enabled: true
        SEARCH_SLOW_LOGS:
          CloudWatchLogsLogGroupArn: "arn:aws:logs:us-east-1:123456789012:log-
group:opensearch-logs"
          Enabled: true
        INDEX_SLOW_LOGS:
          CloudWatchLogsLogGroupArn: "arn:aws:logs:us-east-1:123456789012:log-
group:opensearch-logs"
          Enabled: true
```

Untuk mengetahui detail informasi sintaks, lihat opsi penerbitan log dalam AWS CloudFormation Panduan Pengguna.

Pengaturan ambang OpenSearch logging untuk log lambat

OpenSearchmenonaktifkan log lambat secara default. Setelah Anda mengaktifkan penerbitan log lambatCloudWatch, Anda masih harus menentukan ambang logging untuk setiap indeks. OpenSearch Ambang batas ini menentukan dengan tepat apa yang harus dicatat dalam log dan tingkat log-nya.

Anda menentukan pengaturan ini melalui OpenSearch REST API:

```
PUT domain-endpoint/index/_settings
{
  "index.search.slowlog.threshold.query.warn": "5s",
  "index.search.slowlog.threshold.query.info": "2s"
}
```

Untuk menguji apakah log lambat berhasil melakukan penerbitan, cobalah untuk memulai dengan nilai yang sangat rendah guna memverifikasi bahwa log munculCloudWatch, lalu tingkatkan ambang batas ke tingkat yang lebih berguna.

Jika log tidak muncul, periksa hal berikut ini:

- Apakah grup CloudWatch log ada? Periksa CloudWatch konsol.
- Apakah OpenSearch Layanan memiliki izin untuk menulis ke grup log? Periksa konsol OpenSearch Layanan.
- Apakah domain OpenSearch Layanan dikonfigurasi untuk melakukan penerbitan ke grup log? Periksa konsol OpenSearch Layanan, gunakan AWS CLI `describe-domain-config` opsi, atau hubungi `DescribeDomainConfig` menggunakan salah satu SDK.
- Apakah ambang OpenSearch logging cukup rendah sehingga permintaan Anda melampauinya? Untuk meninjau ambang batas Anda untuk indeks, gunakan perintah berikut:

```
GET domain-endpoint/index/_settings?pretty
```

Jika ingin menonaktifkan log lambat untuk indeks, kembalikan setiap ambang yang Anda ubah ke nilai default mereka yaitu `-1`.

Menonaktifkan penerbitan untuk CloudWatch menggunakan konsol OpenSearch Layanan atau AWS CLI tidak berhenti OpenSearch dari menghasilkan log; itu hanya menghentikan penerbitan log tersebut. Pastikan untuk memeriksa pengaturan indeks jika Anda tidak lagi membutuhkan log lambat.

Melihat log

Melihat aplikasi dan log lambat masuk CloudWatch sama seperti melihat CloudWatch log lainnya. Untuk informasi selengkapnya, lihat Melihat Data Log di Panduan Pengguna Amazon CloudWatch Logs.

Berikut adalah beberapa pertimbangan untuk melihat log:

- OpenSearchLayanan menerbitkan hanya 255.000 karakter pertama dari setiap baris. CloudWatch Konten yang tersisa dipotong. Untuk log audit, tersedia 10.000 karakter per pesan.
- DiCloudWatch, nama aliran log memiliki sufiks`-index-slow-logs`,, `-search-slow-logs-application-logs`, dan `-audit-logs` untuk membantu mengidentifikasi isinya.

Memantau log audit di Amazon OpenSearch Service

Jika domain OpenSearch Layanan Amazon menggunakan kontrol akses berbutir halus, Anda dapat mengaktifkan log audit untuk data Anda. Log audit sangat dapat disesuaikan dan memungkinkan

Anda melacak aktivitas pengguna di OpenSearch klaster Anda, termasuk keberhasilan dan kegagalan otentikasi, permintaan, perubahan indeks OpenSearch, dan kueri penelusuran yang masuk. Konfigurasi default melacak serangkaian tindakan pengguna yang populer, namun sebaiknya sesuaikan pengaturan sesuai kebutuhan Anda.

Sama seperti log OpenSearch aplikasi dan log lambat, OpenSearch Service menerbitkan log audit ke CloudWatch Log. Jika diaktifkan, CloudWatch harga standar berlaku.

> **Note**
>
> Untuk mengaktifkan log audit, peran pengguna Anda harus dipetakan ke peran `security_manager`, yang memberikan Anda akses ke API REST OpenSearch `plugins/_security`. Untuk mempelajari selengkapnya, lihat the section called “Mengubah pengguna utama”.

Topik

Batasan

Log audit memiliki batasan berikut ini:

- Log audit tidak mencakup permintaan pencarian lintas-klaster yang ditolak oleh kebijakan akses domain tujuan.
- Ukuran maksimum setiap pesan log audit adalah 10.000 karakter. Pesan audit log dipotong jika melebihi batas ini.

Mengaktifkan log audit

Mengaktifkan log audit adalah proses dua langkah. Pertama, Anda mengonfigurasi domain Anda untuk mempublikasikan log audit ke CloudWatch Log. Kemudian, Anda mengaktifkan log audit di OpenSearch Dasbor dan mengonfigurasinya untuk memenuhi kebutuhan Anda.

> **Important**
>
> Jika Anda mengalami kesalahan saat mengikuti langkah-langkah ini, lihat the section called "Tidak dapat mengaktifkan log audit" untuk informasi pemecahan masalah.

Langkah 1: Aktifkan log audit dan konfigurasikan kebijakan akses

Langkah-langkah ini menjelaskan cara mengaktifkan log audit menggunakan konsol. Anda juga dapat mengaktifkannya menggunakan AWS CLI, atau API OpenSearch Layanan.

Untuk mengaktifkan log audit untuk domain OpenSearch Layanan (konsol)

1. Pilih domain untuk membuka konfigurasinya, lalu buka tab Log.
2. Pilih log Audit dan kemudian Aktifkan.
3. Buat grup CloudWatch log, atau pilih yang sudah ada.
4. Pilih kebijakan akses yang berisi izin yang sesuai, atau buat kebijakan menggunakan JSON yang disediakan oleh konsol:

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Effect": "Allow",
      "Principal": {
        "Service": "es.amazonaws.com"
      },
      "Action": [
        "logs:PutLogEvents",
        "logs:CreateLogStream"
      ],
      "Resource": "cw_log_group_arn"
    }
  ]
```

```
}
```

Kami menyarankan Anda menambahkan kunci `aws:SourceAccount` dan `aws:SourceArn` kondisi ke kebijakan untuk melindungi diri Anda dari masalah wakil yang membingungkan. Akun sumber adalah pemilik domain dan sumber ARN adalah ARN domain. Domain Anda harus berada di perangkat lunak layanan R20211203 atau yang lebih baru untuk menambahkan kunci kondisi ini.

Misalnya, Anda dapat menambahkan blok kondisi berikut ke kebijakan:

```
"Condition": {
    "StringEquals": {
        "aws:SourceAccount": "account-id"
    },
    "ArnLike": {
        "aws:SourceArn": "arn:aws:es:region:account-id:domain/domain-name"
    }
}
```

5. Pilih Aktifkan.

Langkah 2: Aktifkan log audit di OpenSearch Dasbor

Setelah mengaktifkan log audit di konsol OpenSearch Layanan, Anda juga harus mengaktifkannya di OpenSearch Dasbor dan mengonfigurasinya agar sesuai dengan kebutuhan Anda.

1. Buka OpenSearch Dasbor dan pilih Keamanan dari menu sebelah kiri.
2. Pilih Log Audit.
3. Pilih Mengaktifkan pencatatan audit.

UI Dasbor menawarkan kontrol penuh atas pengaturan log audit di bawah pengaturan Umum dan pengaturan Kepatuhan. Untuk deskripsi semua opsi konfigurasi, lihat Pengaturan log audit.

Aktifkan pencatatan audit menggunakan AWS CLI

AWS CLI Perintah berikut memungkinkan log audit pada domain yang ada:

```
aws opensearch update-domain-config --domain-name my-domain --log-publishing-options
 "AUDIT_LOGS={CloudWatchLogsLogGroupArn=arn:aws:logs:us-east-1:123456789012:log-
group:my-log-group,Enabled=true}"
```

Anda juga dapat mengaktifkan log audit saat membuat domain. Untuk informasi rinci, lihat AWS CLI Referensi Perintah.

Aktifkan pencatatan audit menggunakan API konfigurasi

Permintaan berikut pada API konfigurasi mengaktifkan log audit pada domain yang sudah ada:

```
POST https://es.us-east-1.amazonaws.com/2021-01-01/opensearch/domain/my-domain/config
{
  "LogPublishingOptions": {
    "AUDIT_LOGS": {
      "CloudWatchLogsLogGroupArn":"arn:aws:logs:us-east-1:123456789012:log-
group1:sample-domain",
      "Enabled":true
    }
  }
}
```

Untuk informasi selengkapnya, lihat referensi Amazon OpenSearch Service API.

Lapisan dan kategori log audit

Komunikasi klaster terjadi pada dua lapisan terpisah: lapisan REST dan lapisan transport.

- Lapisan REST mencakup komunikasi dengan klien HTTP seperti curl, Logstash, OpenSearch Dashboards, klien REST tingkat tinggi Java, perpustakaan Permintaan Python — semua permintaan HTTP yang tiba di cluster.
- Lapisan transportasi mencakup komunikasi antara simpul. Misalnya, setelah permintaan pencarian tiba di klaster (atas lapisan REST), simpul koordinasi melayani permintaan mengirimkan kueri ke simpul lain, menerima tanggapan mereka, mengumpulkan dokumen yang diperlukan, dan menyusun mereka ke respon akhir. Operasi seperti alokasi serpihan dan menyeimbangkan ulang juga terjadi di atas lapisan transportasi.

Anda dapat mengaktifkan atau menonaktifkan log audit untuk seluruh lapisan, serta kategori audit individual untuk layer. Tabel berikut berisi ringkasan kategori audit dan lapisan yang tersedia.

Kategori	Deskripsi	Tersedia untuk REST	Tersedia untuk transportasi
FAILED_LOGIN	Permintaan berisi kredensial yang tidak valid, dan autentikasi gagal.	Ya	Ya
MISSING_PRIVILEGES	Pengguna tidak memiliki hak untuk membuat permintaan.	Ya	Ya
GRANTED_PRIVILEGES	Seorang pengguna memiliki hak untuk membuat permintaan.	Ya	Ya
OPENSEARCH_SECURITY_INDEX_ATTEMPT	Permintaan mencoba untuk mengubah indeks `.opendistro_security` .	Tidak	Ya
DIAUTENTIKASI	Permintaan berisi kredensial yang valid, dan autentikasi berhasil.	Ya	Ya
INDEX_EVENT	Permintaan melakukan operasi administratif pada indeks, seperti membuat satu alias, mengatur alias, atau melakukan gabungan kekuatan. Daftar lengkap `indices:admin/` tindakan yang termasuk dalam kategori ini tersedia dalam	Tidak	Ya

Kategori	Deskripsi	Tersedia untuk REST	Tersedia untuk transportasi
	OpenSearch dokumentasi.		

Selain kategori standar ini, kontrol akses detail menawarkan beberapa kategori tambahan yang dirancang untuk memenuhi persyaratan kepatuhan data.

Kategori	Deskripsi
COMPLIANCE_DOC_READ	Permintaan melakukan peristiwa baca pada dokumen dalam indeks.
COMPLIANCE_DOC_WRITE	Permintaan melakukan peristiwa tulis pada dokumen dalam indeks.
COMPLIANCE_INTERNAL_CONFIG_READ	Permintaan melakukan acara baca di indeks `.opendistro_security` .
COMPLIANCE_INTERNAL_CONFIG_WRITE	Permintaan melakukan acara tulis di indeks `.opendistro_security` .

Anda dapat memiliki kombinasi kategori dan atribut pesan mana pun. Misalnya, jika Anda mengirim permintaan REST untuk mengindeks dokumen, Anda mungkin melihat baris berikut dalam log audit:

- AUTHENTICATED pada layer REST (autentikasi)
- GRANTED_PRIVILEGE pada lapisan transport (otorisasi)
- COMPLIANCE_DOC_WRITE (dokumen ditulis ke indeks)

Pengaturan log audit

Log audit memiliki banyak pilihan konfigurasi.

Pengaturan umum

Pengaturan umum memungkinkan Anda mengaktifkan atau menonaktifkan masing-masing kategori atau seluruh lapisan. Kami sangat menyarankan agar meninggalkan GRANTED_PRIVILEGES dan AUTHENTICATED sebagai kategori yang dikecualikan. Jika tidak, kategori ini dicatat untuk setiap permintaan yang valid untuk klaster.

Nama	Pengaturan backend	Deskripsi
Lapisan REST	enable_rest	Mengaktifkan atau menonaktifkan peristiwa yang terjadi pada lapisan REST.
Kategori REST dinonaktifkan	disabled_rest_categories	Tentukan kategori audit untuk diabaikan pada lapisan REST. Memodifikasi kategori ini dapat meningkatkan ukuran log audit secara dramatis.
Lapisan transport	enable_transport	Mengaktifkan atau menonaktifkan peristiwa yang terjadi pada lapisan transport.
Kategori transport dinonaktifkan	disabled_transport_categories	Tentukan kategori audit yang harus diabaikan pada lapisan transport. Memodifikasi kategori ini dapat meningkatkan ukuran log audit secara dramatis.

Pengaturan atribut memungkinkan Anda menyesuaikan jumlah detail di setiap baris log.

Nama	Pengaturan backend	Deskripsi
Permintaan massal	resolve_bulk_requests	Mengaktifkan pengaturan ini menghasilkan log untuk setiap dokumen dalam permintaan massal, yang secara dramatis dapat meningkatkan ukuran log audit.
Isi permintaan	log_request_body	Sertakan badan permintaan pada permintaan.
Selesaikan indeks	resolve_indices	Menyelesaikan alias untuk indeks.

Gunakan pengaturan abaikan untuk mengecualikan satu set pengguna atau jalur API:

Nama	Pengaturan backend	Deskripsi
Pengguna yang diabaikan	ignore_users	Tentukan pengguna yang ingin Anda kecualikan.
Permintaan yang diabaikan	ignore_requests	Tentukan pola permintaan yang ingin Anda kecualikan.

Pengaturan kepatuhan

Setelan kepatuhan memungkinkan Anda menyetel indeks, dokumen, atau akses tingkat bidang.

Nama	Pengaturan backend	Deskripsi
Pencatatan kepatuhan	enable_compliance	Mengaktifkan atau menonaktifkan kepatuhan log.

Anda dapat menentukan pengaturan berikut untuk membaca dan menulis event logging.

Nama	Pengaturan backend	Deskripsi
Pencatatan konfigurasi internal	internal_config	Aktifkan atau nonaktifkan pencatatan peristiwa pada `.opendistro_security` indeks.

Anda dapat menentukan pengaturan berikut untuk peristiwa baca.

Nama	Pengaturan backend	Deskripsi
Metadata baca	read_metadata_only	Sertakan hanya metadata untuk peristiwa baca. Jangan sertakan bidang dokumen apa pun.
Pengguna yang diabaikan	read_ignore_users	Jangan sertakan pengguna tertentu untuk peristiwa baca.
Kolom yang ditonton	read_watched_fields	Tentukan indeks dan bidang untuk menonton peristiwa baca. Menambahkan bidang yang ditonton menghasilkan satu log per akses dokumen, yang dapat meningkatkan ukuran log audit secara dramatis. Bidang yang ditonton mendukung pola indeks dan pola bidang: `{ "index-name-pattern": [ "field-name-pattern" ], "logs*": [ "message" ], "twitter": [ "id", "user*" ] }`

Anda dapat menentukan pengaturan berikut untuk peristiwa tulis.

Nama	Pengaturan backend	Deskripsi
Metadata tulis	write_metadata_only	Sertakan hanya metadata untuk peristiwa tulis. Jangan sertakan bidang dokumen apa pun.
Diffs log	write_log_diffs	Jika write_metadata_only adalah salah, hanya mencakup perbedaan antara peristiwa tulis.

Nama	Pengaturan backend	Deskripsi
Pengguna yang diabaikan	write_ignore_users	Jangan sertakan pengguna tertentu untuk peristiwa tulis.
Tonton indeks	write_watched_indices	Tentukan indeks atau pola indeks untuk ditonton untuk peristiwa tulis. Menambahkan bidang yang ditonton menghasilkan satu log per akses dokumen, yang dapat meningkatkan ukuran log audit secara dramatis.

Contoh log Audit

Bagian ini mencakup contoh konfigurasi, permintaan pencarian, dan log audit yang dihasilkan untuk semua peristiwa baca dan tulis dari sebuah indeks.

Langkah 1: Mengonfigurasi log audit

Setelah mengaktifkan penerbitan log audit ke grup CloudWatch Log, navigasikan ke halaman logging audit OpenSearch Dasbor dan pilih Aktifkan pencatatan audit.

1. Di Pengaturan umum, pilih Konfigurasi dan pastikan bahwa Lapisan REST diaktifkan.
2. Masuk Pengaturan Kepatuhan, pilih Konfigurasi.
3. Pada bagian Tulis, di Bidang yang Ditonton, tambahkan `accounts` untuk semua peristiwa tulis ke indeks ini.
4. Pada bagian Baca, di Bidang yang Ditonton, tambahkan bidang `ssn` dan `id-` pada indeks `accounts`:

```
{
  "accounts-": [
    "ssn",
    "id-"
  ]
}
```

Langkah 2: Lakukan peristiwa baca dan tulis

1. Arahkan ke OpenSearch Dasbor, pilih Alat Pengembang, dan indeks dokumen sampel:

```
PUT accounts/_doc/0
{
  "ssn": "123",
  "id-": "456"
}
```

2. Untuk menguji peristiwa baca, kirim permintaan berikut:

```
GET accounts/_search
{
  "query": {
    "match_all": {}
  }
}
```

Langkah 3: Perhatikan log

1. Buka CloudWatch konsol di https://console.aws.amazon.com/cloudwatch/.
2. Di panel navigasi, pilih Grup log.
3. Pilih grup log yang Anda tentukan saat mengaktifkan log audit. Dalam grup log, OpenSearch Layanan membuat aliran log untuk setiap node di domain Anda.
4. Di Log aliran, pilih Cari semua.
5. Untuk peristiwa baca dan tulis, lihat log terkait. Anda dapat mengharapkan penundaan 5 detik sebelum log muncul.

   Contoh menulis log audit

```
{
  "audit_compliance_operation": "CREATE",
  "audit_cluster_name": "824471164578:audit-test",
  "audit_node_name": "be217225a0b77c2bd76147d3ed3ff83c",
  "audit_category": "COMPLIANCE_DOC_WRITE",
  "audit_request_origin": "REST",
  "audit_compliance_doc_version": 1,
  "audit_node_id": "3xNJhm4XS_yTzEgDWcGRjA",
```

```
  "@timestamp": "2020-08-23T05:28:02.285+00:00",
  "audit_format_version": 4,
  "audit_request_remote_address": "3.236.145.227",
  "audit_trace_doc_id": "lxnJGXQBqZSlDB91r_uZ",
  "audit_request_effective_user": "admin",
  "audit_trace_shard_id": 8,
  "audit_trace_indices": [
    "accounts"
  ],
  "audit_trace_resolved_indices": [
    "accounts"
  ]
}
```

Contoh membaca log audit

```
{
  "audit_cluster_name": "824471164578:audit-docs",
  "audit_node_name": "806f6050cb45437e2401b07534a1452f",
  "audit_category": "COMPLIANCE_DOC_READ",
  "audit_request_origin": "REST",
  "audit_node_id": "saSevm9ASte0-pjAtYi2UA",
  "@timestamp": "2020-08-31T17:57:05.015+00:00",
  "audit_format_version": 4,
  "audit_request_remote_address": "54.240.197.228",
  "audit_trace_doc_id": "config:7.7.0",
  "audit_request_effective_user": "admin",
  "audit_trace_shard_id": 0,
  "audit_trace_indices": [
    "accounts"
  ],
  "audit_trace_resolved_indices": [
    "accounts"
  ]
}
```

Untuk menyertakan badan permintaan, kembali ke pengaturan Kepatuhan di OpenSearch Dasbor dan nonaktifkan metadata Tulis. Untuk mengecualikan peristiwa oleh pengguna tertentu, tambahkan pengguna ke Pengguna yang diabaikan.

Untuk deskripsi setiap bidang log audit, lihat Referensi bidang log audit. Untuk informasi tentang penelusuran dan analisis data log audit Anda, lihat Menganalisis Data CloudWatch Log dengan Wawasan Log di Panduan Pengguna CloudWatch Log Amazon.

Mengonfigurasi log audit menggunakan API REST

Sebaiknya gunakan OpenSearch Dasbor untuk mengonfigurasi log audit, tetapi Anda juga dapat menggunakan REST API kontrol akses berbutir halus. Bagian ini berisi contoh permintaan. Dokumentasi lengkap tentang REST API tersedia dalam OpenSearchdokumentasi.

```
PUT _plugins/_security/api/audit/config
{
  "enabled": true,
  "audit": {
    "enable_rest": true,
    "disabled_rest_categories": [
      "GRANTED_PRIVILEGES",
      "AUTHENTICATED"
    ],
    "enable_transport": true,
    "disabled_transport_categories": [
      "GRANTED_PRIVILEGES",
      "AUTHENTICATED"
    ],
    "resolve_bulk_requests": true,
    "log_request_body": true,
    "resolve_indices": true,
    "exclude_sensitive_headers": true,
    "ignore_users": [
      "kibanaserver"
    ],
    "ignore_requests": [
      "SearchRequest",
      "indices:data/read/*",
      "/_cluster/health"
    ]
  },
  "compliance": {
    "enabled": true,
    "internal_config": true,
    "external_config": false,
    "read_metadata_only": true,
    "read_watched_fields": {
```

```
      "read-index-1": [
        "field-1",
        "field-2"
      ],
      "read-index-2": [
        "field-3"
      ]
    },
    "read_ignore_users": [
      "read-ignore-1"
    ],
    "write_metadata_only": true,
    "write_log_diffs": false,
    "write_watched_indices": [
      "write-index-1",
      "write-index-2",
      "log-*",
      "*"
    ],
    "write_ignore_users": [
      "write-ignore-1"
    ]
  }
}
```

Memantau peristiwa OpenSearch Layanan dengan Amazon EventBridge

OpenSearch Layanan Amazon terintegrasi dengan Amazon EventBridge untuk memberi tahu Anda tentang peristiwa tertentu yang memengaruhi domain Anda. Acara dari AWS layanan dikirimkan ke EventBridge dalam waktu dekat. Acara yang sama juga dikirim ke Amazon CloudWatch Events, pendahulu Amazon EventBridge. Anda dapat menulis aturan sederhana untuk menunjukkan kejadian mana yang sesuai kepentingan Anda, dan tindakan otomatis apa yang diambil ketika suatu kejadian sesuai dengan suatu aturan. Tindakan yang dapat dipicu secara otomatis meliputi hal-hal berikut:

- Memanggil fungsi AWS Lambda
- Melakukan invokasi Amazon EC2 Run Command
- Mengirimkan kejadian ke Amazon Kinesis Data Streams
- Mengaktifkan mesin status AWS Step Functions

- Memberi tahu topik Amazon SNS atau antrean Amazon SQS

Untuk informasi selengkapnya, lihat Memulai Amazon EventBridge di Panduan EventBridge Pengguna Amazon.

Topik

Peristiwa pembaruan perangkat lunak layanan

OpenSearch Layanan mengirimkan peristiwa EventBridge ketika salah satu peristiwa pembaruan perangkat lunak layanan berikut terjadi.

Tersedia pembaruan perangkat lunak layanan

OpenSearch Layanan mengirimkan acara ini ketika pembaruan perangkat lunak layanan tersedia.

Contoh

Berikut adalah contoh peristiwa dari jenis ini:

```
{
  "version": "0",
  "id": "01234567-0123-0123-0123-012345678901",
  "detail-type": "Amazon OpenSearch Service Software Update Notification",
  "source": "aws.es",
  "account": "123456789012",
  "time": "2016-11-01T13:12:22Z",
  "region": "us-east-1",
  "resources": ["arn:aws:es:us-east-1:123456789012:domain/test-domain"],
  "detail": {
```

```
    "event": "Service Software Update",
    "status": "Available",
    "severity": "Informational",
    "description": "Service software update R20220928 available. Service Software
 Deployment Mechanism:
                    Blue/Green. For more information on deployment configuration,
 please
                    see: https://docs.aws.amazon.com/opensearch-service/latest/
developerguide/managedomains-configuration-changes.html"
  }
}
```

Pembaruan perangkat lunak layanan dijadwalkan

OpenSearch Layanan mengirimkan acara ini ketika pembaruan perangkat lunak layanan telah dijadwalkan. Untuk pembaruan opsional, Anda menerima pemberitahuan pada tanggal yang dijadwalkan dan Anda memiliki opsi untuk menjadwal ulang kapan saja. Untuk pembaruan yang diperlukan, Anda menerima pemberitahuan tiga hari sebelum tanggal yang dijadwalkan, dan Anda memiliki opsi untuk menjadwal ulang dalam jendela wajib.

Contoh

Berikut adalah contoh peristiwa dari jenis ini:

```
{
  "version": "0",
  "id": "01234567-0123-0123-0123-012345678901",
  "detail-type": "Amazon OpenSearch Service Software Update Notification",
  "source": "aws.es",
  "account": "123456789012",
  "time": "2016-11-01T13:12:22Z",
  "region": "us-east-1",
  "resources": ["arn:aws:es:us-east-1:123456789012:domain/test-domain"],
  "detail": {
    "event": "Service Software Update",
    "status": "Scheduled",
    "severity": "High",
    "description": "A new service software update [R20200330-p1] has been scheduled at
 [21st May 2023 12:40 GMT].
                    Please see documentation for more information on scheduling
 software updates:
                    https://docs.aws.amazon.com/opensearch-service/latest/
developerguide/service-software.html."
```

```
  }
}
```

Pembaruan perangkat lunak layanan dijadwal ulang

OpenSearch Layanan mengirimkan acara ini ketika pembaruan perangkat lunak layanan opsional telah dijadwalkan ulang. Untuk informasi selengkapnya, lihat the section called “Pembaruan opsional versus yang diperlukan”.

Contoh

Berikut adalah contoh peristiwa dari jenis ini:

```
{
  "version": "0",
  "id": "01234567-0123-0123-0123-012345678901",
  "detail-type": "Amazon OpenSearch Service Software Update Notification",
  "source": "aws.es",
  "account": "123456789012",
  "time": "2016-11-01T13:12:22Z",
  "region": "us-east-1",
  "resources": ["arn:aws:es:us-east-1:123456789012:domain/test-domain"],
  "detail": {
    "event": "Service Software Update",
    "status": "Rescheduled",
    "severity": "High",
    "description": "The service software update [R20200330-p1], which was originally
 scheduled for
                    [21st May 2023 12:40 GMT], has been rescheduled to [23rd May 2023
 12:40 GMT].
                    Please see documentation for more information on scheduling
 software updates:
                    https://docs.aws.amazon.com/opensearch-service/latest/
developerguide/service-software.html."
  }
}
```

Pembaruan perangkat lunak layanan telah selesai

OpenSearch Layanan mengirimkan acara ini ketika pembaruan perangkat lunak layanan telah dimulai.

Contoh

Berikut adalah contoh peristiwa dari jenis ini:

```
{
  "version": "0",
  "id": "01234567-0123-0123-0123-012345678901",
  "detail-type": "Amazon OpenSearch Service Software Update Notification",
  "source": "aws.es",
  "account": "123456789012",
  "time": "2016-11-01T13:12:22Z",
  "region": "us-east-1",
  "resources": ["arn:aws:es:us-east-1:123456789012:domain/test-domain"],
  "detail": {
    "event": "Service Software Update",
    "status": "Started",
    "severity": "Informational",
    "description": "Service software update [R20200330-p1] started.
  }
}
```

Pembaruan perangkat lunak layanan selesai

OpenSearch Layanan mengirimkan acara ini ketika pembaruan perangkat lunak layanan telah selesai.

Contoh

Berikut adalah contoh peristiwa dari jenis ini:

```
{
  "version": "0",
  "id": "01234567-0123-0123-0123-012345678901",
  "detail-type": "Amazon OpenSearch Service Software Update Notification",
  "source": "aws.es",
  "account": "123456789012",
  "time": "2016-11-01T13:12:22Z",
  "region": "us-east-1",
  "resources": ["arn:aws:es:us-east-1:123456789012:domain/test-domain"],
  "detail": {
    "event": "Service Software Update",
    "status": "Completed",
    "severity": "Informational",
    "description": "Service software update [R20200330-p1] completed."
  }
```

```
}
```

Pembaruan perangkat lunak layanan dibatalkan

OpenSearch Layanan mengirimkan acara ini ketika pembaruan perangkat lunak layanan telah dibatalkan.

Contoh

Berikut adalah contoh peristiwa dari jenis ini:

```
{
  "version": "0",
  "id": "01234567-0123-0123-0123-012345678901",
  "detail-type": "Amazon OpenSearch Service Software Update Notification",
  "source": "aws.es",
  "account": "123456789012",
  "time": "2016-11-01T13:12:22Z",
  "region": "us-east-1",
  "resources": ["arn:aws:es:us-east-1:123456789012:domain/test-domain"],
  "detail": {
    "event": "Service Software Update",
    "status": "Cancelled",
    "severity": "Informational",
    "description": "The scheduled service software update [R20200330-p1] has been
 cancelled as a
                    newer update is available. Please schedule the latest update."
  }
}
```

Pembaruan perangkat lunak layanan terjadwal dibatalkan

OpenSearch Layanan mengirimkan acara ini ketika pembaruan perangkat lunak layanan yang sebelumnya dijadwalkan untuk domain telah dibatalkan.

Contoh

Berikut adalah contoh peristiwa dari jenis ini:

```
{
  "version": "0",
```

```
  "id": "01234567-0123-0123-0123-012345678901",
  "detail-type": "Amazon OpenSearch Service Software Update Notification",
  "source": "aws.es",
  "account": "123456789012",
  "time": "2016-11-01T13:12:22Z",
  "region": "us-east-1",
  "resources": ["arn:aws:es:us-east-1:123456789012:domain/test-domain"],
  "detail": {
    "event": "Service Software Update",
    "status": "Cancelled",
    "severity": "Informational",
    "description": "The scheduled service software update [R20200330-p1] has been
 cancelled."
  }
}
```

Pembaruan perangkat lunak layanan tidak dijalankan

OpenSearch Layanan mengirimkan acara ini ketika tidak dapat memulai pembaruan perangkat lunak layanan.

Contoh

Berikut adalah contoh peristiwa dari jenis ini:

```
{
  "version": "0",
  "id": "01234567-0123-0123-0123-012345678901",
  "detail-type": "Amazon OpenSearch Service Software Update Notification",
  "source": "aws.es",
  "account": "123456789012",
  "time": "2016-11-01T13:12:22Z",
  "region": "us-east-1",
  "resources": ["arn:aws:es:us-east-1:123456789012:domain/test-domain"],
  "detail": {
    "event": "Service Software Update",
    "status": "Unexecuted",
    "severity": "Informational",
    "description": "The scheduled service software update [R20200330-p1] cannot be
 started. Reason: [reason]"
  }
}
```

Pembaruan perangkat lunak layanan gagal

OpenSearch Layanan mengirimkan acara ini ketika pembaruan perangkat lunak layanan gagal.

Contoh

Berikut adalah contoh peristiwa dari jenis ini:

```
{
  "version": "0",
  "id": "01234567-0123-0123-0123-012345678901",
  "detail-type": "Amazon OpenSearch Service Software Update Notification",
  "source": "aws.es",
  "account": "123456789012",
  "time": "2016-11-01T13:12:22Z",
  "region": "us-east-1",
  "resources": ["arn:aws:es:us-east-1:123456789012:domain/test-domain"],
  "detail": {
    "event": "Service Software Update",
    "status": "Failed",
    "severity": "High",
    "description": "Installation of service software update [R20200330-p1] failed.
 [reason].
  }
}
```

Pembaruan perangkat lunak layanan diperlukan

OpenSearch Layanan mengirimkan acara ini ketika pembaruan perangkat lunak layanan diperlukan. Untuk informasi selengkapnya, lihat the section called “Pembaruan opsional versus yang diperlukan”.

Contoh

Berikut adalah contoh peristiwa dari jenis ini:

```
{
  "version": "0",
  "id": "01234567-0123-0123-0123-012345678901",
  "detail-type": "Amazon OpenSearch Service Software Update Notification",
  "source": "aws.es",
  "account": "123456789012",
  "time": "2016-11-01T13:12:22Z",
  "region": "us-east-1",
```

```
  "resources": ["arn:aws:es:us-east-1:123456789012:domain/test-domain"],
  "detail": {
    "event": "Service Software Update",
    "status": "Required",
    "severity": "High",
    "description": "Service software update [R20200330-p1] available. Update
                    will be automatically installed after [21st May 2023] if no
                    action is taken. Service Software Deployment Mechanism: Blue/Green.
                    For more information on deployment configuration, please see:
                    https://docs.aws.amazon.com/opensearch-service/latest/
developerguide/managedomains-configuration-changes.html"
  }
}
```

Peristiwa Auto-Tune

OpenSearch Layanan mengirimkan peristiwa ke EventBridge saat salah satu peristiwa Auto-Tune berikut terjadi.

Auto-Tune tertunda

OpenSearch Layanan mengirimkan peristiwa ini ketika Auto-Tune telah mengidentifikasi rekomendasi penyetelan untuk meningkatkan kinerja dan ketersediaan klaster. Anda hanya akan melihat peristiwa ini untuk domain dengan Auto-Tune dinonaktifkan.

Contoh

Berikut adalah contoh peristiwa dari jenis ini:

```
{
  "version": "0",
  "id": "3acb26c8-397c-4c89-a80a-ce672a864c55",
  "detail-type": "Amazon OpenSearch Service Auto-Tune Notification",
  "source": "aws.es",
  "account": "123456789012",
  "time": "2020-10-30T22:06:31Z",
  "region": "us-east-1",
  "resources": ["arn:aws:es:us-east-1:123456789012:domain/test-domain"],
  "detail": {
    "event": "Auto-Tune Event",
    "severity": "Informational",
    "status": "Pending",
```

```
    "description": "Auto-Tune recommends the following new settings for your
 domain: { JVM Heap size : 60%}. Enable Auto-Tune to improve cluster stability and
 performance.",
    "scheduleTime": "{iso8601-timestamp}"
  }
}
```

Auto-Tune dimulai

OpenSearch Layanan mengirimkan acara ini ketika Auto-Tune mulai menerapkan pengaturan baru ke domain Anda.

Contoh

Berikut adalah contoh peristiwa dari jenis ini:

```
{
  "version": "0",
  "id": "3acb26c8-397c-4c89-a80a-ce672a864c55",
  "detail-type": "Amazon OpenSearch Service Auto-Tune Notification",
  "source": "aws.es",
  "account": "123456789012",
  "time": "2020-10-30T22:06:31Z",
  "region": "us-east-1",
  "resources": ["arn:aws:es:us-east-1:123456789012:domain/test-domain"],
  "detail": {
    "event": "Auto-Tune Event",
    "severity": "Informational",
    "status": "Started",
    "scheduleTime": "{iso8601-timestamp}",
    "startTime": "{iso8601-timestamp}",
    "description" : "Auto-Tune is applying the following settings to your domain: { JVM
 Heap size : 60%}."
  }
}
```

Auto-Tune memerlukan deployment biru/hijau terjadwal

OpenSearch Layanan mengirimkan peristiwa ini ketika Auto-Tune telah mengidentifikasi rekomendasi penyetelan yang memerlukan penerapan biru/hijau terjadwal.

Contoh

Berikut adalah contoh peristiwa dari jenis ini:

```
{
  "version": "0",
  "id": "3acb26c8-397c-4c89-a80a-ce672a864c55",
  "detail-type": "Amazon OpenSearch Service Auto-Tune Notification",
  "source": "aws.es",
  "account": "123456789012",
  "time": "2020-10-30T22:06:31Z",
  "region": "us-east-1",
  "resources": ["arn:aws:es:us-east-1:123456789012:domain/test-domain"],
  "detail": {
    "event": "Auto-Tune Event",
    "severity": "Low",
    "status": "Pending",
    "startTime": "{iso8601-timestamp}",
    "description": "Auto-Tune has identified the following settings for your domain
 that require a blue/green deployment: { JVM Heap size : 60%}.
                   You can schedule the deployment for your preferred time."
  }
}
```

Auto-Tune dibatalkan

OpenSearch Layanan mengirimkan acara ini ketika jadwal Auto-Tune telah dibatalkan karena tidak ada rekomendasi tuning yang tertunda.

Contoh

Berikut adalah contoh peristiwa dari jenis ini:

```
{
  "version": "0",
  "id": "3acb26c8-397c-4c89-a80a-ce672a864c55",
  "detail-type": "Amazon OpenSearch Service Auto-Tune Notification",
  "source": "aws.es",
  "account": "123456789012",
  "time": "2020-10-30T22:06:31Z",
  "region": "us-east-1",
  "resources": ["arn:aws:es:us-east-1:123456789012:domain/test-domain"],
  "detail": {
    "event": "Auto-Tune Event",
    "severity": "Low",
```

```
    "status": "Cancelled",
    "scheduleTime": "{iso8601-timestamp}",
    "description": "Auto-Tune has cancelled the upcoming blue/green deployment."
  }
}
```

Auto-Tune selesai

OpenSearch Layanan mengirimkan acara ini ketika Auto-Tune telah menyelesaikan penerapan biru/hijau dan cluster beroperasi dengan pengaturan JVM baru.

Contoh

Berikut adalah contoh peristiwa dari jenis ini:

```
{
  "version": "0",
  "id": "3acb26c8-397c-4c89-a80a-ce672a864c55",
  "detail-type": "Amazon OpenSearch Service Auto-Tune Notification",
  "source": "aws.es",
  "account": "123456789012",
  "time": "2020-10-30T22:06:31Z",
  "region": "us-east-1",
  "resources": ["arn:aws:es:us-east-1:123456789012:domain/test-domain"],
  "detail": {
    "event": "Auto-Tune Event",
    "severity": "Informational",
    "status": "Completed",
    "completionTime": "{iso8601-timestamp}",
    "description": "Auto-Tune has completed the blue/green deployment and successfully
 applied the following settings: { JVM Heap size : 60%}."
  }
}
```

Auto-Tune dinonaktifkan dan perubahan dikembalikan

OpenSearch Layanan mengirimkan acara ini ketika Auto-Tune telah dinonaktifkan dan perubahan yang diterapkan dibatalkan.

Contoh

Berikut adalah contoh peristiwa dari jenis ini:

```
{
  "version": "0",
  "id": "3acb26c8-397c-4c89-a80a-ce672a864c55",
  "detail-type": "Amazon OpenSearch Service Auto-Tune Notification",
  "source": "aws.es",
  "account": "123456789012",
  "time": "2020-10-30T22:06:31Z",
  "region": "us-east-1",
  "resources": [ "arn:aws:es:us-east-1:123456789012:domain/test-domain" ],
  "detail": {
    "event": "Auto-Tune Event",
    "severity": "Informational",
    "status": "Completed",
    "description": "Auto-Tune is now disabled. All settings have been reverted. Auto-
Tune will continue to evaluate
                    cluster performance and provide recommendations.",
    "completionTime": "{iso8601-timestamp}"
  }
}
```

Auto-Tune dinonaktifkan dan perubahan dipertahankan

OpenSearch Layanan mengirimkan acara ini ketika Auto-Tune telah dinonaktifkan dan perubahan yang diterapkan dipertahankan.

Contoh

Berikut adalah contoh peristiwa dari jenis ini:

```
{
  "version": "0",
  "id": "3acb26c8-397c-4c89-a80a-ce672a864c55",
  "detail-type": "Amazon OpenSearch Service Auto-Tune Notification",
  "source": "aws.es",
  "account": "123456789012",
  "time": "2020-10-30T22:06:31Z",
  "region": "us-east-1",
  "resources": ["arn:aws:es:us-east-1:123456789012:domain/test-domain"],
  "detail": {
    "event": "Auto-Tune Event",
    "severity": "Informational",
    "status": "Completed",
```

```
    "description": "Auto-Tune is now disabled. The most-recent settings by Auto-Tune
 have been retained.
                    Auto-Tune will continue to evaluate cluster performance and provide
 recommendations.",
    "completionTime": "{iso8601-timestamp}"
  }
}
```

Acara kesehatan cluster

OpenSearch Layanan mengirimkan peristiwa tertentu ke EventBridge saat kesehatan klaster Anda terganggu.

Pemulihan cluster merah dimulai

OpenSearch Layanan mengirimkan acara ini setelah status klaster Anda terus merah selama lebih dari satu jam. Ini mencoba untuk secara otomatis mengembalikan satu atau lebih indeks merah dari snapshot untuk memperbaiki status cluster.

Contoh

Berikut adalah contoh peristiwa dari jenis ini:

```
{
   "version":"0",
   "id":"01234567-0123-0123-0123-012345678901",
   "detail-type":"Amazon OpenSearch Service Cluster Status Notification",
   "source":"aws.es",
   "account":"123456789012",
   "time":"2016-11-01T13:12:22Z",
   "region":"us-east-1",
   "resources":[
      "arn:aws:es:us-east-1:123456789012:domain/test-domain"
   ],
   "detail":{
      "event":"Automatic Snapshot Restore for Red Indices",
      "status":"Started",
      "severity":"High",
      "description":"Your cluster status is red. We have started automatic snapshot
 restore for the red indices.
                     No action is needed from your side. Red indices [red-index-0, red-
index-1]"
```

```
   }
}
```

Pemulihan cluster merah sebagian selesai

OpenSearch Layanan mengirimkan acara ini ketika hanya dapat mengembalikan subset indeks merah dari snapshot saat mencoba memperbaiki status cluster merah.

Contoh

Berikut adalah contoh peristiwa dari jenis ini:

```
{
   "version":"0",
   "id":"01234567-0123-0123-0123-012345678901",
   "detail-type":"Amazon OpenSearch Service Cluster Status Notification",
   "source":"aws.es",
   "account":"123456789012",
   "time":"2016-11-01T13:12:22Z",
   "region":"us-east-1",
   "resources":[
      "arn:aws:es:us-east-1:123456789012:domain/test-domain"
   ],
   "detail":{
      "event":"Automatic Snapshot Restore for Red Indices",
      "status":"Partially Restored",
      "severity":"High",
      "description":"Your cluster status is red. We were able to restore the following
 Red indices from
                   snapshot: [red-index-0]. Indices not restored: [red-index-1].
 Please refer https://docs.aws.amazon.com/opensearch-service/latest/developerguide/
handling-errors.html#handling-errors-red-cluster-status for troubleshooting steps."
   }
}
```

Pemulihan cluster merah gagal

OpenSearch Layanan mengirimkan acara ini ketika gagal mengembalikan indeks apa pun saat mencoba memperbaiki status cluster merah.

Contoh

Berikut adalah contoh peristiwa dari jenis ini:

```
{
   "version":"0",
   "id":"01234567-0123-0123-0123-012345678901",
   "detail-type":"Amazon OpenSearch Service Cluster Status Notification",
   "source":"aws.es",
   "account":"123456789012",
   "time":"2016-11-01T13:12:22Z",
   "region":"us-east-1",
   "resources":[
      "arn:aws:es:us-east-1:123456789012:domain/test-domain"
   ],
   "detail":{
      "event":"Automatic Snapshot Restore for Red Indices",
      "status":"Failed",
      "severity":"High",
      "description":"Your cluster status is red. We were unable to restore the Red
 indices automatically.
                   Indices not restored: [red-index-0, red-index-1]. Please refer
 https://docs.aws.amazon.com/opensearch-service/latest/developerguide/handling-
errors.html#handling-errors-red-cluster-status for troubleshooting steps."
   }
}
```

Pecahan yang akan dihapus

OpenSearch Layanan mengirimkan acara ini ketika telah mencoba untuk secara otomatis memperbaiki status cluster merah Anda setelah terus merah selama 14 hari, tetapi satu atau lebih indeks tetap merah. Setelah 7 hari lagi (total 21 hari menjadi merah terus menerus), OpenSearch Layanan melanjutkan untuk menghapus pecahan yang tidak ditetapkan pada semua indeks merah.

Contoh

Berikut adalah contoh peristiwa dari jenis ini:

```
{
   "version":"0",
   "id":"01234567-0123-0123-0123-012345678901",
   "detail-type":"Amazon OpenSearch Service Cluster Status Notification",
   "source":"aws.es",
   "account":"123456789012",
   "time":"2022-04-09T10:36:48Z",
   "region":"us-east-1",
```

```
   "resources":[
      "arn:aws:es:us-east-1:123456789012:domain/test-domain"
   ],
   "detail":{
      "severity":"Medium",
      "description":"Your cluster status is red. Please fix the red indices as soon as
 possible.
                    If not fixed by 2022-04-12 01:51:47+00:00, we will delete all
 unassigned shards,
                    the unit of storage and compute, for these red indices to recover
 your domain and make it green.
                    Please refer to https://docs.aws.amazon.com/opensearch-service/
latest/developerguide/handling-errors.html#handling-errors-red-cluster-status for
 troubleshooting steps.
                    test_data, test_data1",
      "event":"Automatic Snapshot Restore for Red Indices",
      "status":"Shard(s) to be deleted"
   }
}
```

Pecahan dihapus

OpenSearch Layanan mengirimkan acara ini setelah status klaster Anda terus merah selama 21 hari. Ini melanjutkan untuk menghapus pecahan yang tidak ditetapkan (penyimpanan dan komputasi) pada semua indeks merah. Untuk detailnya, lihat the section called “Remediasi otomatis cluster merah”.

Contoh

Berikut adalah contoh peristiwa dari jenis ini:

```
{
   "version":"0",
   "id":"01234567-0123-0123-0123-012345678901",
   "detail-type":"Amazon OpenSearch Service Cluster Status Notification",
   "source":"aws.es",
   "account":"123456789012",
   "time":"2022-04-09T10:54:48Z",
   "region":"us-east-1",
   "resources":[
      "arn:aws:es:us-east-1:123456789012:domain/test-domain"
   ],
   "detail":{
```

```
      "severity":"High",
      "description":"We have deleted unassinged shards, the unit of storage and
 compute, in
                    red indices: index-1, index-2 because these indices were red for
 more than
                    21 days and could not be restored with the automated restore
 process.
                    Please refer to https://docs.aws.amazon.com/opensearch-service/
latest/developerguide/handling-errors.html#handling-errors-red-cluster-status for
 troubleshooting steps.",
      "event":"Automatic Snapshot Restore for Red Indices",
      "status":"Shard(s) deleted"
   }
}
```

Peringatan jumlah pecahan tinggi

OpenSearch Layanan mengirimkan peristiwa ini ketika jumlah pecahan rata-rata di seluruh node data panas Anda telah melebihi 90% dari batas default yang direkomendasikan 1.000. Meskipun versi Elasticsearch yang lebih baru dan OpenSearch mendukung jumlah pecahan maks yang dapat dikonfigurasi per batas node, kami sarankan Anda tidak memiliki lebih dari 1.000 pecahan per node. Lihat Memilih jumlah pecahan.

Contoh

Berikut adalah contoh peristiwa dari jenis ini:

```
{
  "version":"0",
  "id":"01234567-0123-0123-0123-012345678901",
  "detail-type":"Amazon OpenSearch Service Notification",
  "source":"aws.es",
  "account":"123456789012",
  "time":"2016-11-01T13:12:22Z",
  "region":"us-east-1",
  "resources":["arn:aws:es:us-east-1:123456789012:domain/test-domain"],
  "detail":{
     "event":"High Shard Count",
     "status":"Warning",
     "severity":"Low",
     "description":"One or more data nodes have close to 1000 shards. To ensure optimum
 performance and stability of your
```

```
                    cluster, please refer to the best practice guidelines - https://
docs.aws.amazon.com/opensearch-service/latest/developerguide/sizing-domains.html#bp-
sharding."
  }
}
```

Batas jumlah pecahan terlampaui

OpenSearch Layanan mengirimkan peristiwa ini ketika jumlah pecahan rata-rata di seluruh node data panas Anda telah melampaui batas default yang disarankan yaitu 1.000. Meskipun versi Elasticsearch yang lebih baru dan OpenSearch mendukung jumlah pecahan maks yang dapat dikonfigurasi per batas node, kami sarankan Anda tidak memiliki lebih dari 1.000 pecahan per node. Lihat Memilih jumlah pecahan.

Contoh

Berikut adalah contoh peristiwa dari jenis ini:

```
{
  "version":"0",
  "id":"01234567-0123-0123-0123-012345678901",
  "detail-type":"Amazon OpenSearch Service Notification",
  "source":"aws.es",
  "account":"123456789012",
  "time":"2016-11-01T13:12:22Z",
  "region":"us-east-1",
  "resources":["arn:aws:es:us-east-1:123456789012:domain/test-domain"],
  "detail":{
     "event":"High Shard Count",
     "status":"Warning",
     "severity":"Medium",
     "description":"One or more data nodes have more than 1000 shards. To ensure
 optimum performance and stability of your
                    cluster, please refer to the best practice guidelines - https://
docs.aws.amazon.com/opensearch-service/latest/developerguide/sizing-domains.html#bp-
sharding."
  }
}
```

Ruang disk rendah

OpenSearch Layanan mengirimkan acara ini ketika satu atau lebih node di cluster Anda memiliki kurang dari 25% ruang penyimpanan yang tersedia, atau kurang dari 25 GB.

Contoh

Berikut adalah contoh peristiwa dari jenis ini:

```
{
  "version":"0",
  "id":"01234567-0123-0123-0123-012345678901",
  "detail-type":"Amazon OpenSearch Service Notification",
  "source":"aws.es",
  "account":"123456789012",
  "time":"2017-12-01T13:12:22Z",
  "region":"us-east-1",
  "resources":["arn:aws:es:us-east-1:123456789012:domain/test-domain"],
  "detail":{
     "event":"Low Disk Space",
     "status":"Warning",
     "severity":"Medium",
     "description":"One or more data nodes in your cluster has less than 25% of storage
 space or less than 25GB.
                  Your cluster will be blocked for writes at 20% or 20GB. Please refer
 to the documentation for more information - https://docs.aws.amazon.com/opensearch-
service/latest/developerguide/handling-errors.html#troubleshooting-cluster-block"
  }
}
```

Pelanggaran watermark disk rendah

OpenSearch Layanan mengirimkan acara ini ketika semua node di cluster Anda memiliki kurang dari 10% ruang penyimpanan yang tersedia, atau kurang dari 10 GB. Ketika semua node melanggar watermark disk rendah, setiap indeks baru menghasilkan cluster kuning, dan ketika semua node jatuh di bawah watermark disk tinggi, itu akan mengarah ke cluster merah.

Contoh

Berikut adalah contoh peristiwa dari jenis ini:

```
{
```

```
  "version":"0",
  "id":"01234567-0123-0123-0123-012345678901",
  "detail-type":"Amazon OpenSearch Service Notification",
  "source":"aws.es",
  "account":"123456789012",
  "time":"2017-12-01T13:12:22Z",
  "region":"us-east-1",
  "resources":["arn:aws:es:us-east-1:123456789012:domain/test-domain"],
  "detail":{
     "event":"Low Disk Watermark Breach",
     "status":"Warning",
     "severity":"Medium",
     "description":"Low Disk Watermark threshold is about to be breached. Once the
 threshold is breached, new index creation will be blocked on all
                   nodes to prevent the cluster status from turning red. Please
 increase disk size to suit your storage needs. For more information,
                   see https://docs.aws.amazon.com/opensearch-service/latest/
developerguide/handling-errors.html#troubleshooting-cluster-block".
  }
}
```

Saldo burst EBS di bawah 70%

OpenSearch Layanan mengirimkan peristiwa ini ketika saldo burst EBS pada satu atau lebih node data turun di bawah 70%. Penipisan keseimbangan burst EBS dapat menyebabkan ketidaktersediaan klaster yang meluas dan pembatasan permintaan I/O, yang dapat menyebabkan latensi dan batas waktu yang tinggi pada permintaan pengindeksan dan pencarian. Untuk langkah-langkah untuk memperbaiki masalah ini, lihatthe section called “Keseimbangan burst EBS rendah”.

Contoh

Berikut adalah contoh peristiwa dari jenis ini:

```
{
  "version":"0",
  "id":"01234567-0123-0123-0123-012345678901",
  "detail-type":"Amazon OpenSearch Service Notification",
  "source":"aws.es",
  "account":"123456789012",
  "time":"2017-12-01T13:12:22Z",
  "region":"us-east-1",
  "resources":["arn:aws:es:us-east-1:123456789012:domain/test-domain"],
```

```
  "detail":{
     "event":"EBS Burst Balance",
     "status":"Warning",
     "severity":"Medium",
     "description":"EBS burst balance on one or more data nodes is below 70%.
                    Follow https://docs.aws.amazon.com/opensearch-service/latest/
developerguide/handling-errors.html#handling-errors-low-ebs-burst
                    to fix this issue."
  }
}
```

Saldo burst EBS di bawah 20%

OpenSearch Layanan mengirimkan peristiwa ini ketika saldo burst EBS pada satu atau lebih node data turun di bawah 20%. Penipisan keseimbangan burst EBS dapat menyebabkan ketidaktersediaan klaster yang meluas dan pembatasan permintaan I/O, yang dapat menyebabkan latensi dan batas waktu yang tinggi pada permintaan pengindeksan dan pencarian. Untuk langkah-langkah untuk memperbaiki masalah ini, lihatthe section called “Keseimbangan burst EBS rendah”.

Contoh

Berikut adalah contoh peristiwa dari jenis ini:

```
{
  "version":"0",
  "id":"01234567-0123-0123-0123-012345678901",
  "detail-type":"Amazon OpenSearch Service Notification",
  "source":"aws.es",
  "account":"123456789012",
  "time":"2017-12-01T13:12:22Z",
  "region":"us-east-1",
  "resources":["arn:aws:es:us-east-1:123456789012:domain/test-domain"],
  "detail":{
     "event":"EBS Burst Balance",
     "status":"Warning",
     "severity":"High",
     "description":"EBS burst balance on one or more data nodes is below 20%.
                    Follow https://docs.aws.amazon.com/opensearch-service/latest/
developerguide/handling-errors.html#handling-errors-low-ebs-burst
                    to fix this issue.
  }
}
```

Throttle throughput disk

OpenSearch Layanan mengirimkan peristiwa ini ketika permintaan baca dan tulis ke domain Anda sedang dibatasi karena keterbatasan throughput volume EBS atau instans EC2 Anda. Jika Anda menerima pemberitahuan ini, pertimbangkan untuk meningkatkan volume atau instans Anda mengikuti praktik terbaik yang AWS disarankan. Jika jenis volume Andagp2, tingkatkan ukuran volume. Jika jenis volume Andagp3, berikan lebih banyak throughput. Anda juga dapat memeriksa bahwa basis instans dan throughput EBS maksimum lebih besar dari atau sama dengan throughput volume yang disediakan, dan dapat meningkatkan skala yang sesuai.

Contoh

Berikut adalah contoh peristiwa dari jenis ini:

```
{
  "version":"0",
  "id":"01234567-0123-0123-0123-012345678901",
  "detail-type":"Amazon OpenSearch Service Notification",
  "source":"aws.es",
  "account":"123456789012",
  "time":"2017-12-01T13:12:22Z",
  "region":"us-east-1",
  "resources":["arn:aws:es:us-east-1:123456789012:domain/test-domain"],
  "detail":{
     "event":"Disk Throughput Throttle",
     "status":"Warning",
     "severity":"Medium",
     "description":"Your domain is experiencing throttling due to instance or volume
 throughput limitations.
                   Please consider scaling your domain to suit your throughput needs.
 In July 2023, we improved
                   the accuracy of throughput throttle calculation by replacing ‘Max
 volume throughput’ with
                   ‘Provisioned volume throughput’. Please refer to the documentation
 for more information."
   }
}
```

Ukuran pecahan besar

OpenSearch Layanan mengirimkan acara ini ketika satu atau beberapa pecahan di cluster Anda telah melampaui 50GiB atau 65GiB. Untuk memastikan kinerja dan stabilitas cluster yang optimal, kurangi ukuran pecahan.

Untuk informasi selengkapnya, lihat praktik terbaik sharding.

Contoh

Berikut adalah contoh peristiwa dari jenis ini:

```
{
  "version":"0",
  "id":"01234567-0123-0123-0123-012345678901",
  "detail-type":"Amazon OpenSearch Service Notification",
  "source":"aws.es",
  "account":"123456789012",
  "time":"2017-12-01T13:12:22Z",
  "region":"us-east-1",
  "resources":["arn:aws:es:us-east-1:123456789012:domain/test-domain"],
  "detail":{
     "event":"Large Shard Size",
     "status":"Warning",
     "severity":"Medium",
     "description":"One or more shards are larger than 65GiB. To ensure optimum cluster
 performance and stability, reduce shard sizes.
                    For more information, see https://docs.aws.amazon.com/opensearch-
service/latest/developerguide/monitoring-events.html#monitoring-events-large-shard-
size."
  }
}
```

Penggunaan JVM yang tinggi

OpenSearch Layanan mengirimkan peristiwa ini ketika `JVMMemoryPressure` metrik untuk domain Anda telah melebihi 80%. Jika melebihi 92% selama 30 menit, semua operasi tulis ke cluster Anda akan diblokir. Untuk memastikan stabilitas klaster yang optimal, kurangi lalu lintas ke klaster atau skala domain Anda untuk menyediakan memori yang cukup untuk beban kerja Anda.

Contoh

Berikut adalah contoh peristiwa dari jenis ini:

```
{
  "version":"0",
  "id":"01234567-0123-0123-0123-012345678901",
  "detail-type":"Amazon OpenSearch Service Notification",
  "source":"aws.es",
  "account":"123456789012",
  "time":"2017-12-01T13:12:22Z",
  "region":"us-east-1",
  "resources":["arn:aws:es:us-east-1:123456789012:domain/test-domain"],
  "detail":{
     "event":"High JVM Usage",
     "status":"Warning",
     "severity":"High",
     "description":"JVM memory pressure has exceeded 80%. If it exceeds 92% for 30
 minutes, all write operations to your cluster
                   will be blocked. To ensure optimum cluster stability, reduce
 traffic to the cluster or use larger instance types.
                   For more information, see https://docs.aws.amazon.com/opensearch-
service/latest/developerguide/monitoring-events.html#monitoring-events-high-jvm."
  }
}
```

GC tidak mencukupi

OpenSearch Layanan mengirimkan acara ini ketika JVM maksimum di atas 70% dan perbedaan antara maksimum dan minimum kurang dari 30%. Ini mungkin menunjukkan bahwa JVM tidak dapat merebut kembali memori yang cukup selama siklus pengumpulan sampah untuk beban kerja Anda. Hal ini dapat menyebabkan respons yang semakin lambat dan latensi yang lebih tinggi; dan dalam beberapa kasus bahkan node turun karena pemeriksaan kesehatan yang habis waktu. Untuk memastikan stabilitas klaster yang optimal, kurangi lalu lintas ke klaster atau skala domain Anda untuk menyediakan memori yang cukup untuk beban kerja Anda.

Contoh

Berikut adalah contoh peristiwa dari jenis ini:

```
{
  "version":"0",
  "id":"01234567-0123-0123-0123-012345678901",
```

```
  "detail-type":"Amazon OpenSearch Service Notification",
  "source":"aws.es",
  "account":"123456789012",
  "time":"2017-12-01T13:12:22Z",
  "region":"us-east-1",
  "resources":["arn:aws:es:us-east-1:123456789012:domain/test-domain"],
  "detail":{
     "event":"Insufficient GC",
     "status":"Warning",
     "severity":"Medium",
     "description":"Maximum JVM is above 70% and JVM range is less than 30%. This may
 indicate insufficient garbage collection for your workload.
                   For more information, see https://docs.aws.amazon.com/opensearch-
service/latest/developerguide/monitoring-events.html#monitoring-events-insufficient-
gc."
  }
}
```

Peringatan perutean indeks khusus

OpenSearch Layanan mengirimkan peristiwa ini ketika domain Anda dalam status pemrosesan dan berisi indeks dengan pengaturan index.routing.allocation khusus yang dapat menyebabkan penerapan biru-hijau macet. Verifikasi pengaturan diterapkan dengan benar.

Contoh

Berikut adalah contoh peristiwa dari jenis ini:

```
{
  "version":"0",
  "id":"01234567-0123-0123-0123-012345678901",
  "detail-type":"Amazon OpenSearch Service Notification",
  "source":"aws.es",
  "account":"123456789012",
  "time":"2017-12-01T13:12:22Z",
  "region":"us-east-1",
  "resources":["arn:aws:es:us-east-1:123456789012:domain/test-domain"],
  "detail":{
     "event":"Custom Index Routing Warning",
     "status":"Warning",
     "severity":"Medium",
     "description":"Your domain is in processing state and contains indice(s) with
 custom index.routing.allocation
```

```
                    settings which can cause blue-green deployments to get stuck.
 Verify settings are applied properly.
                    For more information, see https://docs.aws.amazon.com/opensearch-
service/latest/developerguide/monitoring-events.html#monitoring-events-index-routing."
  }
}
```

Kunci pecahan gagal

OpenSearch Layanan mengirimkan peristiwa ini ketika domain Anda tidak sehat karena pecahan yang tidak ditetapkan dengan. `[ShardLockObtainFailedException]` Untuk informasi selengkapnya, lihat Bagaimana cara mengatasi pengecualian kunci pecahan dalam memori di Amazon Service? OpenSearch

Contoh

Berikut adalah contoh peristiwa dari jenis ini:

```
{
  "version":"0",
  "id":"01234567-0123-0123-0123-012345678901",
  "detail-type":"Amazon OpenSearch Service Notification",
  "source":"aws.es",
  "account":"123456789012",
  "time":"2017-12-01T13:12:22Z",
  "region":"us-east-1",
  "resources":["arn:aws:es:us-east-1:123456789012:domain/test-domain"],
  "detail":{
     "event":"Failed Shard Lock",
     "status":"Warning",
     "severity":"Medium",
     "description":"Your domain is unhealthy due to unassigned shards with
 [ShardLockObtainFailedException]. For more information,
                    see https://docs.aws.amazon.com/opensearch-service/latest/
developerguide/monitoring-events.html#monitoring-events-failed-shard-lock."
}
```

Acara titik akhir VPC

OpenSearch Layanan mengirimkan peristiwa tertentu ke yang EventBridge terkait dengan titik akhir AWS PrivateLink antarmuka.

Pembuatan titik akhir VPC gagal

OpenSearch Layanan mengirimkan acara ini ketika tidak dapat membuat titik akhir VPC yang diminta. Kesalahan ini mungkin terjadi karena Anda telah mencapai batas jumlah endoint VPC yang diizinkan dalam suatu Wilayah. Anda juga akan melihat kesalahan ini jika subnet atau grup keamanan tertentu tidak ada.

Contoh

Berikut adalah contoh peristiwa dari jenis ini:

```
{
   "version":"0",
   "id":"01234567-0123-0123-0123-012345678901",
   "detail-type":"Amazon OpenSearch Service VPC Endpoint Notification",
   "source":"aws.es",
   "account":"123456789012",
   "time":"2016-11-01T13:12:22Z",
   "region":"us-east-1",
   "resources":[
      "arn:aws:es:us-east-1:123456789012:domain/test-domain"
   ],
   "detail":{
      "event":"VPC Endpoint Create Validation",
      "status":"Failed",
      "severity":"High",
      "description":"Unable to create VPC endpoint aos-0d4c74c0342343 for domain
                     arn:aws:es:eu-south-1:123456789012:domain/my-domain due to the
 following validation failures: You've reached the limit on the
                     number of VPC endpoints that you can create in the AWS Region."
   }
}
```

Pembaruan titik akhir VPC gagal

OpenSearch Layanan mengirimkan acara ini ketika tidak dapat menghapus titik akhir VPC yang diminta.

Contoh

Berikut adalah contoh peristiwa dari jenis ini:

```
{
   "version":"0",
   "id":"01234567-0123-0123-0123-012345678901",
   "detail-type":"Amazon OpenSearch Service VPC Endpoint Notification",
   "source":"aws.es",
   "account":"123456789012",
   "time":"2016-11-01T13:12:22Z",
   "region":"us-east-1",
   "resources":[
      "arn:aws:es:us-east-1:123456789012:domain/test-domain"
   ],
   "detail":{
      "event":"VPC Endpoint Update Validation",
      "status":"Failed",
      "severity":"High",
      "description":"Unable to update VPC endpoint aos-0d4c74c0342343 for domain
                     arn:aws:es:eu-south-1:123456789012:domain/my-domain due to the
 following validation failures: <failure message>."
   }
}
```

Penghapusan titik akhir VPC gagal

OpenSearch Layanan mengirimkan acara ini ketika tidak dapat menghapus titik akhir VPC yang diminta.

Contoh

Berikut adalah contoh peristiwa dari jenis ini:

```
{
   "version":"0",
   "id":"01234567-0123-0123-0123-012345678901",
   "detail-type":"Amazon OpenSearch Service VPC Endpoint Notification",
   "source":"aws.es",
   "account":"123456789012",
   "time":"2016-11-01T13:12:22Z",
   "region":"us-east-1",
   "resources":[
      "arn:aws:es:us-east-1:123456789012:domain/test-domain"
   ],
   "detail":{
      "event":"VPC Endpoint Delete Validation",
```

```
      "status":"Failed",
      "severity":"High",
      "description":"Unable to delete VPC endpoint aos-0d4c74c0342343 for domain
                    arn:aws:es:eu-south-1:123456789012:domain/my-domain due to the
 following validation failures: Specified subnet doesn't exist."
   }
}
```

Acara pensiun simpul

OpenSearch Layanan mengirimkan peristiwa EventBridge ketika salah satu peristiwa pensiun node berikut terjadi.

Pensiun node dijadwalkan

OpenSearch Layanan mengirimkan acara ini ketika pensiun node telah dijadwalkan.

Contoh

Berikut adalah contoh peristiwa dari jenis ini:

```
{
  "version": "0",
  "id": "01234567-0123-0123-0123-012345678901",
  "detail-type": "Amazon OpenSearch Service Notification",
  "source": "aws.es",
  "account": "123456789012",
  "time": "2023-04-07T10:07:33Z",
  "region": "us-east-1",
  "resources": ["arn:aws:es:us-east-1:123456789012:domain/test-domain"],
  "detail": {
    "event": "Node Retirement Notification",
    "status": "Scheduled",
    "severity": "Medium",
    "description": "An automated action to retire and replace a node has been scheduled
 on your domain.
                    The node will be replaced in the next off-peak window. For more
 information, see
                    https://docs.aws.amazon.com/opensearch-service/latest/
developerguide/monitoring-events.html."
  }
}
```

Pensiun simpul selesai

OpenSearch Layanan mengirimkan acara ini ketika pensiun node telah selesai.

Contoh

Berikut adalah contoh peristiwa dari jenis ini:

```
{
  "version": "0",
  "id": "01234567-0123-0123-0123-012345678901",
  "detail-type": "Amazon OpenSearch Service Notification",
  "source": "aws.es",
  "account": "123456789012",
  "time": "2023-04-07T10:07:33Z",
  "region": "us-east-1",
  "resources": ["arn:aws:es:us-east-1:123456789012:domain/test-domain"],
  "detail": {
    "event": "Node Retirement Notification",
    "status": "Completed",
    "severity": "Medium",
    "description": "The node has been retired and replaced with a new node."
  }
}
```

Pensiun node gagal

OpenSearch Layanan mengirimkan acara ini ketika pensiun node gagal.

Contoh

Berikut adalah contoh peristiwa dari jenis ini:

```
{
  "version": "0",
  "id": "01234567-0123-0123-0123-012345678901",
  "detail-type": "Amazon OpenSearch Service Notification",
  "source": "aws.es",
  "account": "123456789012",
  "time": "2023-04-07T10:07:33Z",
  "region": "us-east-1",
  "resources": ["arn:aws:es:us-east-1:123456789012:domain/test-domain"],
```

```
  "detail": {
    "event": "Node Retirement Notification",
    "status": "Failed",
    "severity": "Medium",
    "description": "Node retirement failed. No actions are required from your end. We
 will automatically
                    retry replacing the node."
  }
}
```

Peristiwa kesalahan domain

OpenSearch Layanan mengirimkan peristiwa EventBridge ketika salah satu kesalahan domain berikut terjadi.

Kegagalan validasi pembaruan domain

OpenSearch Layanan mengirimkan peristiwa ini jika mengalami satu atau beberapa kegagalan validasi saat mencoba memperbarui atau melakukan perubahan konfigurasi pada domain. Untuk langkah-langkah untuk mengatasi kegagalan ini, lihatthe section called “Memecahkan masalah kesalahan validasi”.

Contoh

Berikut adalah contoh peristiwa dari jenis ini:

```
{
   "version":"0",
   "id":"01234567-0123-0123-0123-012345678901",
   "detail-type":"Amazon OpenSearch Service Domain Update Notification",
   "source":"aws.es",
   "account":"123456789012",
   "time":"2016-11-01T13:12:22Z",
   "region":"us-east-1",
   "resources":[
      "arn:aws:es:us-east-1:123456789012:domain/test-domain"
   ],
   "detail":{
      "event":"Domain Update Validation",
      "status":"Failed",
      "severity":"High",
```

```
      "description":"Unable to perform updates to your domain due to the following
 validation failures: <failures>
                    Please see the documentation for more information https://
docs.aws.amazon.com/opensearch-service/latest/developerguide/managedomains-
configuration-changes.html#validation"
   }
}
```

Kunci KMS tidak dapat diakses

OpenSearch Layanan mengirimkan acara ini ketika tidak dapat mengakses AWS KMS kunci Anda.

Contoh

Berikut adalah contoh peristiwa dari jenis ini:

```
{
  "version":"0",
  "id":"01234567-0123-0123-0123-012345678901",
  "detail-type":"Domain Error Notification",
  "source":"aws.es",
  "account":"123456789012",
  "time":"2016-11-01T13:12:22Z",
  "region":"us-east-1",
  "resources":["arn:aws:es:us-east-1:123456789012:domain/test-domain"],
  "detail":{
     "event":"KMS Key Inaccessible",
     "status":"Error",
     "severity":"High",
     "description":"The KMS key associated with this domain is inaccessible. You are at
 risk of losing access to your domain.
                    For more information, please refer to https://docs.aws.amazon.com/
opensearch-service/latest/developerguide/encryption-at-rest.html#disabled-key."
  }
}
```

Isolasi domain

OpenSearch Layanan mengirimkan peristiwa ini ketika domain Anda menjadi terisolasi dan tidak dapat menerima, membaca, atau menulis permintaan karena tidak dapat dijangkau oleh jaringan.

Contoh

Berikut adalah contoh peristiwa dari jenis ini:

```
{
  "version":"0",
  "id":"01234567-0123-0123-0123-012345678901",
  "detail-type":"Amazon OpenSearch Service Notification",
  "source":"aws.es",
  "account":"123456789012",
  "time":"2023-11-01T13:12:22Z",
  "region":"us-east-1",
  "resources":["arn:aws:es:us-east-1:123456789012:domain/test-domain"],
  "detail":{
     "event":"Domain Isolation Notification",
     "status":"Error",
     "severity":"High",
     "description":"Your OpenSearch Service domain has been isolated. An isolated
 domain is unreachable by network and cannot receive, read, or write requests. For more
 information and assistance, please contact AWS Support at https://docs.aws.amazon.com/
opensearch-service/latest/developerguide/encryption-at-rest.html#disabled-key."
  }
}
```

Tutorial: Mendengarkan EventBridge acara OpenSearch Layanan Amazon

Dalam tutorial ini, Anda menyiapkan AWS Lambda fungsi sederhana yang mendengarkan peristiwa Amazon OpenSearch Service dan menuliskannya ke aliran CloudWatch log Log.

Prasyarat

Tutorial ini mengasumsikan bahwa Anda memiliki domain OpenSearch Layanan yang ada. Jika Anda belum membuat domain, ikuti langkah-langkah di Membuat dan mengelola domain untuk membuat domain.

Langkah 1: Buat fungsi Lambda

Dalam prosedur ini, Anda membuat fungsi Lambda sederhana untuk berfungsi sebagai target pesan acara OpenSearch Layanan.

Untuk membuat fungsi Lambda target

1. Buka AWS Lambda konsol di https://console.aws.amazon.com/lambda/.
2. Pilih Buat fungsi dan Penulis dari awal.

3. Untuk nama Fungsi, masukkan event-handler.
4. Untuk Waktu pengoperasian, pilih Python 3.8.
5. Pilih Buat fungsi.
6. Di bagian Kode fungsi, edit kode sampel untuk mencocokkan contoh berikut:

```
import json

def lambda_handler(event, context):
    if event["source"] != "aws.es":
        raise ValueError("Function only supports input from events with a source
 type of: aws.es")

    print(json.dumps(event))
```

   Ini adalah fungsi Python 3.8 sederhana yang mencetak peristiwa yang dikirim oleh Layanan. OpenSearch Jika semuanya dikonfigurasi dengan benar, di akhir tutorial ini, detail peristiwa muncul di aliran CloudWatch log Log yang terkait dengan fungsi Lambda ini.
7. Pilih Terapkan.

Langkah 2: Mendaftarkan aturan peristiwa

Pada langkah ini, Anda membuat EventBridge aturan yang menangkap peristiwa dari domain OpenSearch Layanan Anda. Aturan ini menangkap semua peristiwa dalam akun yang didefinisikan. Pesan peristiwa itu sendiri berisi informasi tentang sumber peristiwa, termasuk domain dari mana ia berasal. Anda dapat menggunakan informasi ini untuk mem-filter dan mengurutkan peristiwa secara terprogram.

Untuk membuat EventBridge aturan

1. Buka EventBridge konsol di https://console.aws.amazon.com/events/.
2. Pilih Buat aturan.
3. Sebutkan aturan aturan acara.
4. Pilih Berikutnya.
5. Untuk pola acara, pilih AWS layanan, OpenSearch Layanan Amazon, dan Semua Acara. Pola ini berlaku di semua domain OpenSearch Layanan Anda dan untuk setiap peristiwa OpenSearch Layanan. Atau, Anda dapat membuat pola yang lebih spesifik untuk mem-filter beberapa hasil.
6. Tekan Berikutnya.

7. Untuk target, pilih fungsi Lambda. Di dropdown fungsi, pilih event-handler.
8. Tekan Berikutnya.
9. Lewati tag dan tekan Berikutnya lagi.
10. Tinjau konfigurasi dan pilih Buat aturan.

Langkah 3: Uji konfigurasi Anda

Lain kali Anda menerima pemberitahuan di bagian Pemberitahuan konsol OpenSearch Layanan, jika semuanya dikonfigurasi dengan benar, fungsi Lambda Anda dipicu dan menulis data peristiwa ke aliran CloudWatch log Log untuk fungsi tersebut.

Untuk menguji konfigurasi Anda

1. Buka CloudWatch konsol di https://console.aws.amazon.com/cloudwatch/.
2. Pada panel navigasi, pilih Log dan pilih grup log untuk fungsi Lambda Anda (misalnya, /aws/lambda/event-handler).
3. Pilih stream log untuk melihat data peristiwa.

Tutorial: Mengirim peringatan Amazon SNS untuk pembaruan perangkat lunak yang tersedia

Dalam tutorial ini, Anda mengonfigurasi aturan EventBridge acara Amazon yang menangkap pemberitahuan untuk pembaruan perangkat lunak layanan yang tersedia di OpenSearch Layanan Amazon dan mengirimi Anda pemberitahuan email melalui Amazon Simple Notification Service (Amazon SNS).

Prasyarat

Tutorial ini mengasumsikan bahwa Anda memiliki domain OpenSearch Layanan yang ada. Jika Anda belum membuat domain, ikuti langkah-langkah di Membuat dan mengelola domain untuk membuat domain.

Langkah 1: Buat dan berlangganan ke topik Amazon SNS

Mengonfigurasi topik Amazon SNS untuk melayani sebagai target peristiwa untuk aturan perisitwa baru Anda.

Untuk membuat target Amazon SNS

1. Buka konsol Amazon SNS di https://console.aws.amazon.com/sns/v3/home.
2. Pilih Topik dan Buat topik.
3. Untuk jenis pekerjaan, pilih Standar, dan beri nama pembaruan perangkat lunak pekerjaan.
4. Pilih Buat topik.
5. Setelah topik dibuat, pilih Buat langganan.
6. Untuk Protokol, pilih Email. UntukTitik akhir, masukkan alamat email yang Anda dapat mengaksesnya dan pilih Buat langganan.
7. Periksa akun email Anda dan tunggu untuk menerima pesan email konfirmasi langganan. Saat Anda menerimanya, pilih Konfirmasi langganan.

Langkah 2: Mendaftarkan aturan peristiwa

Selanjutnya, daftarkan aturan peristiwa yang hanya merekam peristiwa pembaruan perangkat lunak layanan.

Untuk membuat aturan acara

1. Buka EventBridge konsol di https://console.aws.amazon.com/events/.
2. Pilih Buat aturan.
3. Beri nama aturan softwareupdate-rule.
4. Pilih Berikutnya.
5. Untuk pola acara, pilih AWS layanan, OpenSearch Layanan Amazon, dan Pemberitahuan Pembaruan Perangkat Lunak OpenSearch Layanan Amazon. Pola ini cocok dengan setiap peristiwa pembaruan perangkat lunak layanan dari OpenSearch Layanan. Untuk informasi selengkapnya tentang pola peristiwa, lihat pola EventBridge peristiwa Amazon di Panduan EventBridge Pengguna Amazon.
6. Secara opsional, Anda dapat memfilter hanya untuk tingkat keparahan tertentu. Untuk tingkat keparahan setiap acara, lihatthe section called “Peristiwa pembaruan perangkat lunak layanan”.
7. Pilih Berikutnya.
8. Untuk target, pilih topik SNS dan pilih pembaruan perangkat lunak.
9. Pilih Berikutnya.

10. Lewati tag dan pilih Berikutnya.
11. Tinjau konfigurasi aturan dan pilih Buat aturan.

Lain kali Anda menerima pemberitahuan dari OpenSearch Layanan tentang pembaruan perangkat lunak layanan yang tersedia, jika semuanya dikonfigurasi dengan benar, Amazon SNS akan mengirimi Anda peringatan email tentang pembaruan.

Pemantauan panggilan API Amazon OpenSearch Service dengan AWS CloudTrail

Amazon OpenSearch Service terintegrasi denganAWS CloudTrail, yaitu sebuah layanan yang menyediakan catatan tindakan yang dilakukan oleh pengguna, peran, atau AWS layanan di OpenSearch Layanan. CloudTrailmenangkap semua panggilan API konfigurasi untuk OpenSearch Service sebagai peristiwa.

> **Note**
>
> CloudTrailhanya menangkap panggilan ke Configuration API, seperti `CreateDomain` dan`GetUpgradeStatus`. CloudTrailtidak menangkap panggilan ke OpenSearchAPI, seperti `_search` dan`_bulk`. Untuk panggilan ini, lihat the section called “Memantau log audit”.

Panggilan yang direkam mencakup panggilan dari konsolOpenSearch,,AWS CLI, atau AWS SDK. Jika Anda membuat jejak, Anda dapat mengaktifkan pengiriman berkelanjutan dari CloudTrail kejadian ke bucket Amazon S3, termasuk peristiwa untuk OpenSearch Service. Jika Anda tidak mengonfigurasi jejak, Anda masih dapat melihat kejadian terbaru di CloudTrail konsol di Riwayat peristiwa. Dengan menggunakan informasi yang dikumpulkan olehCloudTrail, Anda dapat menentukan permintaan yang dibuat ke OpenSearch Service, alamat IP asal permintaan tersebut dibuat, siapa yang membuat permintaan, kapan permintaan dibuat, dan detail lainnya.

Untuk mempelajari lebih lanjutCloudTrail, lihat Panduan AWS CloudTrail Pengguna.

Informasi OpenSearch Layanan Amazon di CloudTrail

CloudTraildiaktifkan pada Anda Akun AWS saat Anda membuat akun. Ketika aktivitas terjadi di OpenSearch Service, aktivitas tersebut dicatat dalam CloudTrail peristiwa bersama peristiwa AWS layanan lainnya di Riwayat peristiwa. Anda dapat melihat, mencari, dan mengunduh peristiwa terbaru

di akun Akun AWS Anda. Untuk informasi lebih lanjut, lihat Menampilkan peristiwa dengan riwayat peristiwa CloudTrail.

Untuk catatan berkelanjutan tentang peristiwa di Akun AWS akun Anda, termasuk peristiwa untuk OpenSearch Service, buat jejak. Jejak memungkinkan CloudTrail untuk mengirim berkas log ke bucket Amazon S3. Secara default, saat Anda membuat jejak di konsol, jejak tersebut berlaku untuk semua Wilayah AWS. Jejak mencatat peristiwa dari semua Wilayah di partisi AWS dan mengirimkan berkas log ke bucket Amazon S3 yang Anda tentukan. Selain itu, Anda dapat membuat konfigurasi layanan AWS lainnya untuk menganalisis lebih lanjut dan bertindak berdasarkan data peristiwa yang dikumpulkan di log CloudTrail. Untuk informasi selengkapnya, lihat yang berikut:

- Membuat jejak untuk Anda Akun AWS
- AWSintegrasi layanan dengan Log CloudTrail
- Mengonfigurasi Amazon SNS untuk CloudTrail
- Menerima berkas log CloudTrail dari beberapa wilayah dan Menerima berkas log CloudTrail dari beberapa akun

Semua tindakan API konfigurasi OpenSearch Layanan dicatat oleh CloudTrail dan didokumentasikan dalam Amazon OpenSearch Service API Reference.

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 AWS Identity and Access Management (IAM)
- Jika permintaan tersebut dibuat dengan kredensial keamanan sementara untuk peran atau pengguna gabungan
- Jika permintaan tersebut dibuat oleh layanan AWS lainnya

Untuk informasi lain, lihat Elemen userIdentity CloudTrail.

Memahami entri file log Amazon OpenSearch Service

Jejak adalah konfigurasi yang memungkinkan pengiriman peristiwa sebagai berkas log ke bucket Amazon S3 yang telah Anda tentukan. Berkas log CloudTrail berisi satu atau beberapa entri log. Peristuwa mewakili satu permintaan dari sumber apa pun dan mencakup informasi tentang tindakan yang diminta, tanggal dan waktu tindakan, parameter permintaan, dan sebagainya. Berkas log

CloudTrail bukan jejak tumpukan terurut dari panggilan API publik, sehingga berkas tersebut tidak muncul dalam urutan tertentu.

Contoh berikut menunjukkan entri CloudTrail log yang menunjukkan `CreateDomain` operasi:


```
{
  "eventVersion": "1.05",
  "userIdentity": {
    "type": "IAMUser",
    "principalId": "AIDACKCEVSQ6C2EXAMPLE",
    "arn": "arn:aws:iam::123456789012:user/test-user",
    "accountId": "123456789012",
    "accessKeyId": "access-key",
    "userName": "test-user",
    "sessionContext": {
      "attributes": {
        "mfaAuthenticated": "false",
        "creationDate": "2018-08-21T21:59:11Z"
      }
    },
    "invokedBy": "signin.amazonaws.com"
  },
  "eventTime": "2018-08-21T22:00:05Z",
  "eventSource": "es.amazonaws.com",
  "eventName": "CreateDomain",
  "awsRegion": "us-west-1",
  "sourceIPAddress": "123.123.123.123",
  "userAgent": "signin.amazonaws.com",
  "requestParameters": {
    "engineVersion": "OpenSearch_1.0",
    "clusterConfig": {
      "instanceType": "m4.large.search",
      "instanceCount": 1
    },
    "snapshotOptions": {
      "automatedSnapshotStartHour": 0
    },
    "domainName": "test-domain",
    "encryptionAtRestOptions": {},
    "eBSOptions": {
      "eBSEnabled": true,
      "volumeSize": 10,
      "volumeType": "gp2"
    },
```

```
    "accessPolicies": "{\"Version\":\"2012-10-17\",\"Statement\":[{\"Effect\":\"Allow
\",\"Principal\":{\"AWS\":[\"123456789012\"]},\"Action\":[\"es:*\"],\"Resource\":
\"arn:aws:es:us-west-1:123456789012:domain/test-domain/*\"}]}",
    "advancedOptions": {
      "rest.action.multi.allow_explicit_index": "true"
    }
  },
  "responseElements": {
    "domainStatus": {
      "created": true,
      "clusterConfig": {
        "zoneAwarenessEnabled": false,
        "instanceType": "m4.large.search",
        "dedicatedMasterEnabled": false,
        "instanceCount": 1
      },
      "cognitoOptions": {
        "enabled": false
      },
      "encryptionAtRestOptions": {
        "enabled": false
      },
      "advancedOptions": {
        "rest.action.multi.allow_explicit_index": "true"
      },
      "upgradeProcessing": false,
      "snapshotOptions": {
        "automatedSnapshotStartHour": 0
      },
      "eBSOptions": {
        "eBSEnabled": true,
        "volumeSize": 10,
        "volumeType": "gp2"
      },
      "engineVersion": "OpenSearch_1.0",
      "processing": true,
      "aRN": "arn:aws:es:us-west-1:123456789012:domain/test-domain",
      "domainId": "123456789012/test-domain",
      "deleted": false,
      "domainName": "test-domain",
      "accessPolicies": "{\"Version\":\"2012-10-17\",\"Statement\":[{\"Effect\":\"Allow
\",\"Principal\":{\"AWS\":\"arn:aws:iam::123456789012:root\"},\"Action\":\"es:*\",
\"Resource\":\"arn:aws:es:us-west-1:123456789012:domain/test-domain/*\"}]}"
    }
```

```
  },
  "requestID": "12345678-1234-1234-1234-987654321098",
  "eventID": "87654321-4321-4321-4321-987654321098",
  "eventType": "AwsApiCall",
  "recipientAccountId": "123456789012"
}
```

Keamanan di OpenSearch Layanan Amazon

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

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

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

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

Topik

- Mengonfigurasi otentikasi Amazon Cognito untuk Dasbor OpenSearch
- Menggunakan peran terkait layanan untuk Amazon Service OpenSearch

Perlindungan data di Amazon OpenSearch Service

Model tanggung jawab AWS bersama model berlaku untuk perlindungan data di OpenSearch Layanan Amazon. Seperti yang dijelaskan dalam model AWS ini, bertanggung jawab untuk melindungi infrastruktur global yang menjalankan semua AWS Cloud. Anda bertanggung jawab untuk mempertahankan kendali atas konten yang di-host pada infrastruktur ini. Anda juga bertanggung jawab atas tugas-tugas konfigurasi dan manajemen keamanan untuk Layanan AWS yang Anda gunakan. Lihat informasi yang lebih lengkap tentang privasi data dalam Pertanyaan Umum Privasi Data. Lihat informasi tentang perlindungan data di Eropa di pos blog Model Tanggung Jawab Bersama dan GDPR AWS di Blog Keamanan AWS .

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

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

Kami sangat merekomendasikan agar Anda tidak pernah memasukkan informasi identifikasi yang sensitif, seperti nomor rekening pelanggan Anda, ke dalam tanda atau bidang isian bebas seperti bidang Nama. Ini termasuk saat Anda bekerja dengan OpenSearch Service atau lainnya Layanan

AWS menggunakan konsol, API AWS CLI, atau AWS SDK. Data apa pun yang Anda masukkan ke dalam tanda atau bidang isian bebas yang digunakan untuk nama dapat digunakan untuk log penagihan atau log diagnostik. Saat Anda memberikan URL ke server eksternal, kami sangat menganjurkan supaya Anda tidak menyertakan informasi kredensial di dalam URL untuk memvalidasi permintaan Anda ke server itu.

Enkripsi data saat istirahat untuk OpenSearch Layanan Amazon

OpenSearch Domain layanan menawarkan enkripsi data saat istirahat, fitur keamanan yang membantu mencegah akses tidak sah ke data Anda. Fitur ini menggunakan AWS Key Management Service (AWS KMS) untuk menyimpan dan mengelola kunci enkripsi Anda dan algoritma Advanced Encryption Standard dengan kunci 256-bit (AES-256) untuk melakukan enkripsi. Jika diaktifkan, fitur mengenkripsi aspek-aspek domain berikut:

- Semua indeks (termasuk yang ada di UltraWarm penyimpanan)
- OpenSearch log
- Swap file
- Semua data lain dalam direktori aplikasi
- Snapshot otomatis

Berikut ini tidak dienkripsi saat Anda mengaktifkan enkripsi data saat tidak digunakan, namun Anda dapat mengambil langkah tambahan untuk melindunginya:

- Snapshot manual: Saat ini Anda tidak dapat menggunakan AWS KMS kunci untuk mengenkripsi snapshot manual. Namun, Anda dapat menggunakan enkripsi sisi server dengan kunci yang dikelola S3 atau kunci KMS untuk mengenkripsi bucket yang Anda gunakan sebagai repositori snapshot. Untuk petunjuk, lihat the section called “Mendaftarkan repositori snapshot manual”.
- Log lambat dan log kesalahan: Jika Anda mempublikasikan log dan ingin mengenkripsinya, Anda dapat mengenkripsi grup CloudWatch log Log mereka menggunakan AWS KMS kunci yang sama dengan domain OpenSearch Layanan. Untuk informasi selengkapnya, lihat Mengenkripsi data CloudWatch log di Log menggunakan AWS KMS Panduan Pengguna Amazon CloudWatch Logs.

> **Note**
>
> Anda tidak dapat mengaktifkan enkripsi saat istirahat di domain yang ada jika UltraWarm atau penyimpanan dingin diaktifkan di domain. Anda harus terlebih dahulu menonaktifkan

UltraWarm atau penyimpanan dingin, mengaktifkan enkripsi saat istirahat, dan kemudian mengaktifkan kembali UltraWarm atau penyimpanan dingin. Jika Anda ingin menyimpan indeks di dalam UltraWarm atau penyimpanan dingin, Anda harus memindahkannya ke penyimpanan panas sebelum menonaktifkan UltraWarm atau penyimpanan dingin.

OpenSearch Layanan hanya mendukung kunci KMS enkripsi simetris, bukan kunci asimetris. Untuk mempelajari cara membuat kunci simetris, lihat Membuat kunci di Panduan AWS Key Management Service Pengembang.

Terlepas dari apakah enkripsi saat istirahat diaktifkan, semua domain secara otomatis mengenkripsi paket khusus menggunakan AES-256 dan kunci yang dikelola Layanan. OpenSearch

Izin

Untuk menggunakan konsol OpenSearch Layanan untuk mengonfigurasi enkripsi data saat istirahat, Anda harus memiliki izin membaca AWS KMS, seperti kebijakan berbasis identitas berikut:

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

Jika Anda ingin menggunakan kunci selain kunci yang AWS dimiliki, Anda juga harus memiliki izin untuk membuat hibah untuk kunci tersebut. Izin ini biasanya mengambil bentuk kebijakan berbasis sumber daya yang Anda tentukan ketika Anda membuat kunci.

Jika Anda ingin menjaga kunci Anda tetap eksklusif untuk OpenSearch Layanan, Anda dapat menambahkan ViaService kondisi kms: ke kebijakan kunci tersebut:

```
"Condition": {
  "StringEquals": {
```

```
    "kms:ViaService": "es.us-west-1.amazonaws.com"
  },
  "Bool": {
    "kms:GrantIsForAWSResource": "true"
  }
}
```

Untuk informasi selengkapnya, lihat Menggunakan kebijakan utama di AWS KMS di Panduan AWS Key Management Service Pengembang.

Mengaktifkan enkripsi data saat tidak digunakan

Enkripsi data saat istirahat pada domain baru memerlukan salah satu OpenSearch atau Elasticsearch 5.1 atau yang lebih baru. Mengaktifkannya di domain yang ada memerlukan salah satu OpenSearch atau Elasticsearch 6.7 atau yang lebih baru.

Untuk mengaktifkan enkripsi data saat istirahat (konsol)

1. Buka domain di AWS konsol, lalu pilih Tindakan dan Edit konfigurasi keamanan.
2. Di bawah Enkripsi, pilih Aktifkan enkripsi data saat istirahat.
3. Pilih AWS KMS kunci yang akan digunakan, lalu pilih Simpan perubahan.

Anda juga dapat mengaktifkan enkripsi melalui API konfigurasi. Permintaan berikut memungkinkan enkripsi data saat istirahat pada domain yang ada:

```
{
   "ClusterConfig":{
      "EncryptionAtRestOptions":{
         "Enabled": true,
         "KmsKeyId":"arn:aws:kms:us-east-1:123456789012:alias/my-key"
      }
   }
}
```

Kunci KMS dinonaktifkan atau dihapus

Jika Anda menonaktifkan atau menghapus kunci yang Anda gunakan untuk mengenkripsi domain, domain menjadi tidak dapat diakses. OpenSearch Layanan mengirimi Anda pemberitahuan yang memberi tahu Anda bahwa itu tidak dapat mengakses kunci KMS. Aktifkan kembali kunci segera untuk mengakses domain Anda.

Tim OpenSearch Layanan tidak dapat membantu memulihkan data Anda jika kunci Anda dihapus. AWS KMS menghapus kunci hanya setelah masa tunggu setidaknya tujuh hari. Jika kunci Anda tertunda penghapusan, batalkan penghapusan atau ambil snapshot manual domain untuk mencegah hilangnya data.

Menonaktifkan enkripsi data saat tidak digunakan

Setelah mengonfigurasi domain untuk mengenkripsi data saat tidak digunakan, Anda tidak dapat menonaktifkan pengaturan. Sebagai gantinya, Anda dapat mengambil Snapshot manual dari domain yang sudah ada, membuat domain lain, memigrasikan data Anda, dan menghapus domain lama.

Memantau domain yang mengenkripsi data saat tidak digunakan

Domain yang mengenkripsi data saat tidak digunakan memiliki dua metrik tambahan: `KMSKeyError` dan `KMSKeyInaccessible`. Metrik ini hanya muncul jika domain mengalami masalah dengan kunci enkripsi Anda. Untuk deskripsi lengkap metrik ini, lihat the section called “Metrik klaster”. Anda dapat melihatnya menggunakan konsol OpenSearch Layanan atau CloudWatch konsol Amazon.

> **Tip**
>
> Setiap metrik mewakili masalah yang signifikan untuk domain, jadi sebaiknya Anda membuat CloudWatch alarm untuk keduanya. Untuk informasi selengkapnya, lihat the section called “ CloudWatch Alarm yang direkomendasikan”.

Pertimbangan lainnya

- Rotasi kunci otomatis mempertahankan properti AWS KMS kunci Anda, sehingga rotasi tidak berpengaruh pada kemampuan Anda untuk mengakses OpenSearch data Anda. Domain OpenSearch Layanan Terenkripsi tidak mendukung rotasi kunci manual, yang melibatkan pembuatan kunci baru dan memperbarui referensi apa pun ke kunci lama. Untuk mempelajari selengkapnya, lihat Memutar kunci di Panduan AWS Key Management Service Pengembang.
- Tipe instans tertentu tidak mendukung enkripsi data saat tidak digunakan. Untuk rincian selengkapnya, lihat the section called “Tipe instans yang didukung”.
- Domain yang mengenkripsi data saat tidak digunakan menggunakan nama repositori yang berbeda untuk snapshot otomatis mereka. Untuk informasi selengkapnya, lihat the section called “Memulihkan snapshot”.

- Meskipun kami sangat menyarankan untuk mengaktifkan enkripsi saat istirahat, ini dapat menambahkan overhead CPU tambahan dan latensi beberapa milidetik. Namun, sebagian besar kasus penggunaan tidak sensitif terhadap perbedaan ini, dan besarnya dampaknya bergantung pada konfigurasi klaster, klien, dan profil penggunaan Anda.

ode-to-node Enkripsi N untuk OpenSearch Layanan Amazon

ode-to-node Enkripsi N menyediakan lapisan keamanan tambahan di atas fitur default OpenSearch Layanan Amazon.

Setiap domain OpenSearch Layanan — terlepas dari apakah domain tersebut menggunakan akses VPC — berada dalam VPC khusus miliknya sendiri. Arsitektur ini mencegah penyerang potensial mencegat lalu lintas antar OpenSearch node dan menjaga cluster tetap aman. Secara default, bagaimanapun, lalu lintas dalam VPC tidak dienkripsi. ode-to-node Enkripsi N memungkinkan enkripsi TLS 1.2 untuk semua komunikasi dalam VPC.

Jika Anda mengirim data ke OpenSearch Layanan melalui HTTPS, node-to-node enkripsi membantu memastikan bahwa data Anda tetap dienkripsi sebagai OpenSearch mendistribusikan (dan mendistribusikan ulang) ke seluruh cluster. Jika data tiba tanpa dienkripsi melalui HTTP, OpenSearch Layanan mengenkripsi setelah mencapai cluster. Anda dapat mengharuskan semua lalu lintas ke domain tiba melalui HTTPS menggunakan konsol, AWS CLI, atau API konfigurasi.

ode-to-node Enkripsi N diperlukan jika Anda mengaktifkan kontrol akses berbutir halus.

Mengaktifkan enkripsi node-to-node

ode-to-node Enkripsi N pada domain baru memerlukan versi apa pun dari OpenSearch, atau Elasticsearch 6.0 atau yang lebih baru. Mengaktifkan node-to-node enkripsi pada domain yang ada memerlukan versi apa pun dari OpenSearch, atau Elasticsearch 6.7 atau yang lebih baru. Pilih domain yang ada di konfigurasi keamanan AWS konsol, Tindakan, dan Edit.

Atau, Anda dapat menggunakan API konfigurasi AWS CLI atau. Untuk informasi selengkapnya, lihat Referensi AWS CLI Perintah dan Referensi API OpenSearch Layanan.

Menonaktifkan enkripsi node-to-node

Setelah Anda mengonfigurasi domain untuk menggunakan node-to-node enkripsi, Anda tidak dapat menonaktifkan pengaturan. Sebagai gantinya, Anda dapat mengambil Snapshot manual dari domain yang sudah dienkripsi, membuat domain lain, memigrasikan data Anda, dan menghapus domain lama.

Identity and Access Management di Amazon OpenSearch Service

Amazon OpenSearch Service menawarkan beberapa cara untuk mengontrol akses ke domain Anda. Topik ini mencakup berbagai tipe kebijakan, bagaimana kebijakan berinteraksi satu sama lain, dan cara membuat kebijakan kustom Anda sendiri.

> **Important**
>
> Dukungan VPC memperkenalkan beberapa pertimbangan tambahan untuk OpenSearch kontrol akses Layanan. Untuk informasi selengkapnya, lihat the section called “Tentang kebijakan akses pada domain VPC”.

Tipe kebijakan

OpenSearch Layanan mendukung tiga jenis kebijakan akses:

- the section called “Kebijakan berbasis sumber daya”
- the section called “Kebijakan berbasis identitas”
- the section called “Kebijakan berbasis IP”

Kebijakan berbasis sumber daya

Anda menambahkan kebijakan berbasis sumber daya, sering disebut kebijakan akses domain, ketika Anda membuat domain. Kebijakan ini menentukan tindakan mana yang dapat dilakukan prinsipal pada subsumber daya domain (dengan pengecualian pencarian lintas klaster). Subresource termasuk OpenSearch indeks dan API. Elemen Utama menentukan akun, pengguna, atau peran yang diizinkan akses. Elemen Sumber Daya menentukan sub sumber daya mana utama ini dapat akses.

Misalnya, kebijakan berbasis sumber daya berikut memberikan akses `test-user` penuh (`es:*`) ke subsumber daya pada: `test-domain`

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

```
        "AWS": [
          "arn:aws:iam::123456789012:user/test-user"
        ]
      },
      "Action": [
        "es:*"
      ],
      "Resource": "arn:aws:es:us-west-1:987654321098:domain/test-domain/*"
    }
  ]
}
```

Dua pertimbangan penting berlaku untuk kebijakan ini:

- Hak istimewa ini hanya berlaku untuk domain ini. Kecuali Anda membuat kebijakan serupa di domain lain, `test-user` hanya dapat mengakses `test-domain`.
- Jejak `/*` di elemen `Resource` adalah signifikan dan menunjukkan bahwa kebijakan berbasis sumber daya hanya berlaku untuk sub sumber daya domain, bukan domain itu sendiri. Dalam kebijakan berbasis sumber daya, aksi `es:*` setara dengan `es:ESHttp*`.

  Misalnya, `test-user` dapat membuat permintaan terhadap indeks (`GET https://search-test-domain.us-west-1.es.amazonaws.com/test-index`), namun tidak dapat memperbarui konfigurasi domain (`POST https://es.us-west-1.amazonaws.com/2021-01-01/opensearch/domain/test-domain/config`). Perhatikan perbedaan antara dua titik akhir. Mengakses API konfigurasi memerlukan kebijakan berbasis identitas.

Anda dapat menentukan nama indeks sebagian dengan menambahkan wildcard. Contoh ini mengidentifikasi indeks apa pun yang dimulai dengan: `commerce`

```
arn:aws:es:us-west-1:987654321098:domain/test-domain/commerce*
```

Dalam hal ini, wildcard berarti `test-user` dapat membuat permintaan ke indeks di dalamnya `test-domain` yang memiliki nama yang dimulai. `commerce`

Untuk lebih membatasi `test-user`, Anda dapat menerapkan kebijakan berikut:

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

```
    {
      "Effect": "Allow",
      "Principal": {
        "AWS": [
          "arn:aws:iam::123456789012:user/test-user"
        ]
      },
      "Action": [
        "es:ESHttpGet"
      ],
      "Resource": "arn:aws:es:us-west-1:987654321098:domain/test-domain/commerce-data/
_search"
    }
  ]
}
```

Sekarang hanya `test-user` dapat melakukan satu operasi: pencarian terhadap `commerce-data` indeks. Semua indeks lain dalam domain tidak dapat diakses, dan tanpa izin untuk menggunakan `es:ESHttpPut` atau `es:ESHttpPost` tindakan, tidak `test-user` dapat menambahkan atau memodifikasi dokumen.

Selanjutnya, Anda dapat memutuskan untuk mengonfigurasi peran bagi pengguna daya. Kebijakan ini memberikan akses `power-user-role` ke HTTP GET dan metode PUT untuk semua URI dalam indeks:

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Effect": "Allow",
      "Principal": {
        "AWS": [
          "arn:aws:iam::123456789012:role/power-user-role"
        ]
      },
      "Action": [
        "es:ESHttpGet",
        "es:ESHttpPut"
      ],
      "Resource": "arn:aws:es:us-west-1:987654321098:domain/test-domain/commerce-data/
*"
    }
  ]
```

```
}
```

Jika domain Anda berada di VPC atau menggunakan kontrol akses detail, Anda dapat menggunakan kebijakan akses domain terbuka. Jika tidak, kebijakan akses domain Anda harus berisi beberapa pembatasan, baik berdasarkan prinsip atau alamat IP.

Untuk informasi tentang semua tindakan yang tersedia, lihat the section called “Referensi elemen kebijakan”. Untuk kontrol yang jauh lebih rinci atas data Anda, gunakan kebijakan akses domain terbuka dengan kontrol akses detail.

Kebijakan berbasis identitas

Tidak seperti kebijakan berbasis sumber daya, yang merupakan bagian dari setiap domain OpenSearch Layanan, Anda melampirkan kebijakan berbasis identitas kepada pengguna atau peran yang menggunakan layanan (IAM). AWS Identity and Access Management Sama seperti kebijakan berbasis sumber daya, kebijakan berbasis identitas menentukan siapa yang dapat mengakses layanan, tindakan mana yang dapat mereka lakukan, dan jika berlaku, sumber daya tempat mereka dapat melakukan tindakan tersebut.

Meskipun kebijakan tentu tidak harus, kebijakan berbasis identitas cenderung lebih generik. Kebijakan sering kali hanya mengatur tindakan API konfigurasi yang dapat dilakukan pengguna. Setelah kebijakan ini diterapkan, Anda dapat menggunakan kebijakan berbasis sumber daya (atau kontrol akses berbutir halus) di OpenSearch Layanan untuk menawarkan akses kepada pengguna ke indeks dan API. OpenSearch

> **Note**
>
> Pengguna dengan `AmazonOpenSearchServiceReadOnlyAccess` kebijakan AWS terkelola tidak dapat melihat status kesehatan klaster di konsol. Untuk memungkinkan mereka melihat status kesehatan klaster (dan OpenSearch data lainnya), tambahkan `es:ESHttpGet` tindakan ke kebijakan akses dan lampirkan ke akun atau peran mereka.

Karena kebijakan berbasis identitas melekat pada pengguna atau peran (utama), JSON tidak menentukan yang utama. Kebijakan berikut memberikan akses ke tindakan yang dimulai dengan `Describe` dan `List`. Kombinasi tindakan ini menyediakan akses hanya-baca ke konfigurasi domain, tetapi tidak untuk data yang disimpan dalam domain itu sendiri:

```
{
```

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

Administrator mungkin memiliki akses penuh ke OpenSearch Layanan dan semua data yang disimpan di semua domain:

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

Kebijakan berbasis identitas memungkinkan Anda menggunakan tag untuk mengontrol akses ke API konfigurasi. Kebijakan berikut, misalnya, memungkinkan utama yang terlampir melihat dan memperbarui konfigurasi domain jika domain memiliki tanda `team:devops`:

```
{
  "Version": "2012-10-17",
  "Statement": [{
    "Action": [
      "es:UpdateDomainConfig",
      "es:DescribeDomain",
      "es:DescribeDomainConfig"
    ],
    "Effect": "Allow",
```

```
    "Resource": "*",
    "Condition": {
      "ForAnyValue:StringEquals": {
        "aws:ResourceTag/team": [
          "devops"
        ]
      }
    }
  }]
}
```

Anda juga dapat menggunakan tag untuk mengontrol akses ke OpenSearch API. Kebijakan berbasis tag untuk OpenSearch API hanya berlaku untuk metode HTTP. Misalnya, kebijakan berikut memungkinkan prinsipal terlampir mengirim permintaan GET dan PUT ke OpenSearch API jika domain memiliki tag: `environment:production`

```
{
  "Version": "2012-10-17",
  "Statement": [{
    "Action": [
      "es:ESHttpGet",
      "es:ESHttpPut"
    ],
    "Effect": "Allow",
    "Resource": "*",
    "Condition": {
      "ForAnyValue:StringEquals": {
        "aws:ResourceTag/environment": [
          "production"
        ]
      }
    }
  }]
}
```

Untuk kontrol OpenSearch API yang lebih terperinci, pertimbangkan untuk menggunakan kontrol akses berbutir halus.

> **Note**
>
> Setelah menambahkan satu atau beberapa OpenSearch API ke kebijakan berbasis tag, Anda harus melakukan operasi tag tunggal (seperti menambahkan, menghapus, atau memodifikasi

tag) agar perubahan diterapkan pada domain. Anda harus menggunakan perangkat lunak layanan R20211203 atau yang lebih baru untuk menyertakan operasi OpenSearch API dalam kebijakan berbasis tag.

OpenSearch Layanan mendukung `RequestTag` dan kunci kondisi `TagKeys` global untuk API konfigurasi, bukan OpenSearch API. Kondisi ini hanya berlaku untuk panggilan API yang menyertakan tanda dalam permintaan, seperti `CreateDomain`, `AddTags`, dan `RemoveTags`. Kebijakan berikut memungkinkan prinsipal terlampir membuat domain, tetapi hanya jika kebijakan menyertakan tanda `team:it` dalam permintaan:

```
{
  "Version": "2012-10-17",
  "Statement": {
    "Effect": "Allow",
    "Action": [
      "es:CreateDomain",
      "es:AddTags"
    ],
    "Resource": "*",
    "Condition": {
      "StringEquals": {
        "aws:RequestTag/team": [
          "it"
        ]
      }
    }
  }
}
```

Untuk rincian lebih lanjut tentang menggunakan tanda untuk kontrol akses dan perbedaan antara kebijakan berbasis sumber daya dan kebijakan berbasis identitas, lihat Panduan Pengguna IAM.

Kebijakan berbasis IP

Kebijakan berbasis IP membatasi akses ke domain ke satu atau lebih alamat IP atau blok CIDR. Secara teknis, kebijakan berbasis IP bukanlah jenis kebijakan yang berbeda. Sebaliknya, mereka hanya kebijakan berbasis sumber daya yang menentukan prinsipal anonim dan menyertakan elemen Ketentuan khusus.

Daya tarik utama kebijakan berbasis IP adalah bahwa mereka mengizinkan permintaan yang tidak ditandatangani ke domain OpenSearch Layanan, yang memungkinkan Anda menggunakan klien seperti curl dan OpenSearch Dasbor atau mengakses domain melalui server proxy. Untuk mempelajari informasi lebih lanjut, lihat the section called “Menggunakan proxy untuk mengakses OpenSearch Layanan dari OpenSearch Dasbor”.

> **Note**
>
> Jika Anda mengaktifkan akses VPC untuk domain, Anda tidak dapat mengonfigurasi kebijakan berbasis IP. Sebagai gantinya, Anda bisa menggunakan grup keamanan untuk mengontrol alamat IP yang dapat mengakses domain. Untuk informasi selengkapnya, lihat the section called “Tentang kebijakan akses pada domain VPC”.

Kebijakan berikut memberikan semua permintaan HTTP yang berasal dari akses kisaran IP tertentu ke `test-domain`:

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Effect": "Allow",
      "Principal": {
        "AWS": "*"
      },
      "Action": [
        "es:ESHttp*"
      ],
      "Condition": {
        "IpAddress": {
          "aws:SourceIp": [
            "192.0.2.0/24"
          ]
        }
      },
      "Resource": "arn:aws:es:us-west-1:987654321098:domain/test-domain/*"
    }
  ]
}
```

Jika domain Anda memiliki titik akhir publik dan tidak menggunakan kontrol akses detail, sebaiknya gabungkan prinsip IAM dan alamat IP. Kebijakan ini memberikan akses HTTP `test-user` hanya jika permintaan berasal dari kisaran IP yang ditentukan:

```
{
  "Version": "2012-10-17",
  "Statement": [{
    "Effect": "Allow",
    "Principal": {
      "AWS": [
        "arn:aws:iam::987654321098:user/test-user"
      ]
    },
    "Action": [
      "es:ESHttp*"
    ],
    "Condition": {
      "IpAddress": {
        "aws:SourceIp": [
          "192.0.2.0/24"
        ]
      }
    },
    "Resource": "arn:aws:es:us-west-1:987654321098:domain/test-domain/*"
  }]
}
```

Membuat dan menandatangani Permintaan OpenSearch layanan

Meskipun Anda mengonfigurasi kebijakan akses berbasis sumber daya yang sepenuhnya terbuka, semua permintaan ke API konfigurasi OpenSearch Layanan harus ditandatangani. Jika kebijakan Anda menentukan peran IAM atau pengguna, permintaan ke OpenSearch API juga harus ditandatangani menggunakan Versi AWS Tanda Tangan 4. Metode penandatanganan berbeda dengan API:

- Untuk melakukan panggilan ke API konfigurasi OpenSearch Layanan, kami sarankan Anda menggunakan salah satu AWS SDK. SDK sangat menyederhanakan proses dan dapat menghemat banyak waktu dibandingkan dengan membuat dan menandatangani permintaan Anda sendiri. Titik akhir API konfigurasi menggunakan format berikut:

```
es.region.amazonaws.com/2021-01-01/
```

Misalnya, permintaan berikut membuat perubahan konfigurasi ke domain `movies`, tetapi Anda harus menandatanganinya sendiri (tidak disarankan):

```
POST https://es.us-east-1.amazonaws.com/2021-01-01/opensearch/domain/movies/config
{
  "ClusterConfig": {
    "InstanceType": "c5.xlarge.search"
  }
}
```

Jika Anda menggunakan salah satu SDK, seperti Boto 3, SDK secara otomatis menangani penandatanganan permintaan:

```
import boto3

client = boto3.client(es)
response = client.update_domain_config(
  DomainName='movies',
  ClusterConfig={
    'InstanceType': 'c5.xlarge.search'
  }
)
```

Untuk contoh kode Java, lihat the section called “Menggunakan AWS SDKs”.

- Untuk melakukan panggilan ke OpenSearch API, Anda harus menandatangani permintaan Anda sendiri. OpenSearch API menggunakan format berikut:

```
domain-id.region.es.amazonaws.com
```

Sebagai contoh, permintaan berikut mencari indeks `movies` untuk thor:

```
GET https://my-domain.us-east-1.es.amazonaws.com/movies/_search?q=thor
```

> **Note**
>
> Layanan mengabaikan parameter yang dikirimkan di URL untuk permintaan HTTP POST yang ditandatangani dengan Tanda Tangan Versi 4.

Ketika kebijakan bertabrakan

Kompleksitas muncul ketika kebijakan tidak setuju atau tidak menyebutkan secara eksplisit pengguna. Memahami cara kerja IAM dalam Panduan Pengguna IAM memberikan ringkasan singkat logika evaluasi kebijakan:

- Secara default, semua permintaan ditolak.
- Izin secara eksplisit akan mengabaikan default ini.
- Penolakan secara eksplisit akan mengabaikan izin apa pun.

Misalnya, jika kebijakan berbasis sumber daya memberi Anda akses ke subresource domain (OpenSearch indeks atau API), tetapi kebijakan berbasis identitas menolak akses Anda, Anda ditolak aksesnya. Jika kebijakan berbasis identitas memberikan akses dan kebijakan berbasis sumber daya tidak menentukan apakah Anda harus memiliki akses atau tidak, Anda diperbolehkan akses. Lihat tabel kebijakan berpotongan berikut untuk ringkasan lengkap hasil untuk subsumber daya domain.

	Diizinkan dalam kebijakan berbasis sumber daya	Ditolak dalam kebijakan berbasis sumber daya	Tidak diperbolehkan atau ditolak dalam kebijakan berbasis sumber daya
Allowed in identity-based policy	Izinkan	Deny	Allow
Denied in identity-based policy	Deny	Deny	Deny
Neither allowed nor denied in identity-based policy	Allow	Deny	Deny

Referensi elemen kebijakan

OpenSearch Layanan mendukung sebagian besar elemen kebijakan dalam Referensi Elemen Kebijakan IAM, dengan pengecualian. `NotPrincipal` Tabel berikut menunjukkan elemen yang paling umum.

Elemen kebijakan JSON	Ringkasan
`Version`	Versi bahasa kebijakan saat ini adalah `2012-10-17` . Semua kebijakan akses harus menentukan nilai ini.
`Effect`	Elemen ini menentukan apakah pernyataan memungkinkan atau menolak akses ke tindakan yang ditentukan. Nilai yang benar adalah `Allow` atau `Deny`.
`Principal`	Elemen ini menentukan peran Akun AWS atau IAM atau pengguna yang diizinkan atau ditolak akses ke sumber daya dan dapat mengambil beberapa bentuk: • AWS akun: `"Principal":{"AWS": ["123456789012"]}` atau `"Principal":{"AWS": ["arn:aws:iam::123456789012:root"]}` • Pengguna IAM: `"Principal":{"AWS": ["arn:aws:iam::123456789012:user/test-user"]}` • Peran IAM: `"Principal":{"AWS": ["arn:aws:iam::123456789012:role/test-role"]}` **Important** Menentukan `*` karakter pengganti akan mengaktifkan akses anonim ke domain, yang tidak kami sarankan kecuali Anda menambahkan kondisi berbasis IP, menggunakan dukungan VPC, atau mengaktifkan kontrol akses detail. Selain itu, periksa dengan cermat kebijakan berikut untuk mengonfirmasi bahwa kebijakan tersebut tidak memberikan akses luas:

Elemen kebijakan JSON	Ringkasan
	• Kebijakan berbasis identitas yang dilampirkan pada AWS prins ipal terkait (misalnya, peran IAM) • Kebijakan berbasis sumber daya yang dilampirkan pada AWS sumber daya terkait (misalnya, kunci KMS) AWS Key Management Service

Elemen kebijakan JSON	Ringkasan

`Action`	OpenSearch Layanan menggunakan `ESHttp*` tindakan untuk metode OpenSearch HTTP. Tindakan lainnya berlaku untuk API konfigurasi.
	
	Tindakan `es:` tertentu mendukung izin tingkat sumber daya. Misalnya, Anda dapat memberikan izin pengguna untuk menghapus satu domain tertentu tanpa memberikan izin pengguna untuk menghapus domain mana pun. Tindakan lain hanya berlaku untuk layanan itu sendiri. Misalnya,`es:ListDomainNames` tidak masuk akal dalam konteks domain tunggal dan dengan demikian memerlukan wildcard.
	
	Untuk daftar semua tindakan yang tersedia dan apakah itu berlaku untuk subresource domain (`test-domain/*`), ke konfigurasi domain (`test-domain`), atau hanya untuk layanan (`*`), lihat Tindakan, sumber daya, dan kunci kondisi untuk OpenSearch Layanan Amazon di Referensi Otorisasi Layanan
	Kebijakan Berbasis sumber daya berbeda dari izin tingkat sumber daya. Kebijakan berbasis sumber daya adalah kebijakan JSON penuh yang melekat pada domain. Izin tingkat sumber daya memungkinkan Anda membatasi tindakan untuk domain atau subsumber daya tertentu. Dalam praktiknya, Anda dapat memikirkan izin tingkat sumber daya sebagai bagian opsional dari sumber daya atau kebijakan berbasis identitas.
	
	Meskipun izin tingkat sumber daya untuk `es:CreateDomain` mungkin tampak tidak intuitif—bagaimanapun juga, mengapa memberikan izin kepada pengguna untuk membuat domain yang sudah ada?—peng gunaan karakter pengganti memungkinkan Anda menerapkan skema penamaan sederhana untuk domain Anda, seperti `"Resource": "arn:aws:es:us-west-1:987654321098:domain/my-team-name-*"` .
	
	Tentu saja, tidak ada yang mencegah Anda memasukkan tindakan bersama elemen sumber daya yang kurang ketat, seperti berikut ini:
	
	`{`
	`"Version": "2012-10-17",`

Elemen kebijakan JSON	Ringkasan

	<pre>"Statement": [
	  {
	  "Effect": "Allow",
	  "Action": [
	  "es:ESHttpGet",
	  "es:DescribeDomain"
	  ],
	  "Resource": "*"
	  }
	  ]
	}</pre>
	Untuk mempelajari selengkapnya tentang memasangkan tindakan dan sumber daya, lihat elemen `Resource` dalam tabel ini.

`Condition`	OpenSearch Layanan mendukung sebagian besar kondisi yang dijelaskan dalam kunci konteks kondisi AWS global di Panduan Pengguna IAM. Pengecualian penting termasuk `aws:PrincipalTag` kunci, yang OpenSearch Layanan tidak mendukung.
	
	Saat mengonfigurasi kebijakan berbasis IP, Anda menentukan alamat IP atau blok CIDR sebagai kondisi, seperti berikut:
	
	<pre>"Condition": {
	  "IpAddress": {
	  "aws:SourceIp": [
	  "192.0.2.0/32"
	  ]
	  }
	}</pre>
	Seperti disebutkan dalam the section called “Kebijakan berbasis identitas ”`aws:ResourceTag` , kunci`aws:RequestTag` ,, dan `aws:TagKeys` kondisi berlaku untuk API konfigurasi serta OpenSearch API.

Elemen kebijakan JSON	Ringkasan

`Resource`	OpenSearch Layanan menggunakan `Resource` elemen dalam tiga cara dasar:
	
	• Untuk tindakan yang berlaku untuk OpenSearch Layanan itu sendiri, seperties:`ListDomainNames` , atau untuk mengizinkan akses penuh, gunakan sintaks berikut:
	
	`"Resource": "*"`
	
	• Untuk tindakan yang melibatkan konfigurasi domain, seperti `es:DescribeDomain` , Anda dapat menggunakan sintaks berikut:
	
	`"Resource": "arn:aws:es:` *region*`:`*aws-account-id*`:domain/`*domain-name*` "`
	
	• Untuk tindakan yang diterapkan ke domain subsumber daya, seperti `es:ESHttpGet` , Anda dapat menggunakan sintaks berikut:
	
	`"Resource": "arn:aws:es:` *region*`:`*aws-account-id*`:domain/`*domain-name*` /*"`
	
	Anda tidak perlu menggunakan wildcard. OpenSearch Layanan memungkinkan Anda menentukan kebijakan akses yang berbeda untuk setiap OpenSearch indeks atau API. Misalnya, Anda dapat membatasi izin pengguna ke indeks `test-index` :
	
	`"Resource": "arn:aws:es:` *region*`:`*aws-account-id*`:domain/`*domain-name*` /test-index"`
	
	Alih-alih akses penuh ke `test-index` , Anda mungkin lebih memilih untuk membatasi kebijakan hanya pada API pencarian:
	
	`"Resource": "arn:aws:es:` *region*`:`*aws-account-id*`:domain/`*domain-name*` /test-index/_search"`

Elemen kebijakan JSON	Ringkasan
	Anda bahkan dapat mengontrol akses ke dokumen individual: `"Resource": "arn:aws:es:  region:aws-account-id:domain/domain-name /test-index/test-type/1"` Pada dasarnya, jika OpenSearch mengekspresikan subresource sebagai URI, Anda dapat mengontrol akses ke sana menggunakan kebijakan akses. Untuk kontrol yang lebih besar atas sumber daya yang dapat diakses oleh pengguna, lihat the section called “Kontrol akses detail”. Untuk rincian tentang tindakan mana yang mendukung izin tingkat sumber daya, lihat elemen `Action` di tabel ini.

Pilihan lanjutan dan pertimbangan API

OpenSearch Layanan memiliki beberapa opsi lanjutan, salah satunya memiliki implikasi kontrol akses:`rest.action.multi.allow_explicit_index`. Pada pengaturan default benar, memungkinkan pengguna untuk memotong izin subsumber daya dalam keadaan tertentu.

Misalnya, pertimbangkan kebijakan berbasis sumber daya berikut:

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Effect": "Allow",
      "Principal": {
        "AWS": [
          "arn:aws:iam::123456789012:user/test-user"
        ]
      },
      "Action": [
        "es:ESHttp*"
      ],
```

```
      "Resource": [
        "arn:aws:es:us-west-1:987654321098:domain/test-domain/test-index/*",
        "arn:aws:es:us-west-1:987654321098:domain/test-domain/_bulk"
      ]
    },
    {
      "Effect": "Allow",
      "Principal": {
        "AWS": [
          "arn:aws:iam::123456789012:user/test-user"
        ]
      },
      "Action": [
        "es:ESHttpGet"
      ],
      "Resource": "arn:aws:es:us-west-1:987654321098:domain/test-domain/restricted-
index/*"
    }
  ]
}
```

Kebijakan ini memberikan akses `test-user` penuh ke `test-index` dan API OpenSearch massal. Hal ini juga memungkinkan permintaan `GET` untuk `restricted-index`.

Permintaan pengindeksan berikut, seperti yang Anda harapkan, gagal karena kesalahan izin:

```
PUT https://search-test-domain.us-west-1.es.amazonaws.com/restricted-index/movie/1
{
  "title": "Your Name",
  "director": "Makoto Shinkai",
  "year": "2016"
}
```

Berbeda dengan API indeks, API massal memungkinkan Anda membuat, memperbarui, dan menghapus banyak dokumen dalam satu panggilan. Anda sering menentukan operasi ini di isi permintaan, namun, bukan di URL permintaan. Karena OpenSearch Layanan menggunakan URL untuk mengontrol akses ke subresource domain, pada kenyataannya, `test-user` dapat menggunakan API massal untuk membuat perubahan. `restricted-index` Meskipun pengguna tidak memiliki izin `POST` pada indeks, permintaan berikut berhasil:

```
POST https://search-test-domain.us-west-1.es.amazonaws.com/_bulk
{ "index" : { "_index": "restricted-index", "_type" : "movie", "_id" : "1" } }
```

```
{ "title": "Your Name", "director": "Makoto Shinkai", "year": "2016" }
```

Dalam situasi ini, kebijakan akses gagal untuk memenuhi tujuannya. Untuk mencegah pengguna melewati pembatasan semacam ini, Anda dapat mengubah `rest.action.multi.allow_explicit_index` ke salah. Jika nilai ini salah, semua panggilan ke sebagian API besar, mget, dan msearch yang menentukan nama indeks dalam isi permintaan berhenti bekerja. Dengan kata lain, panggilan ke `_bulk` tidak lagi bekerja, tapi panggilan ke `test-index/_bulk` bekerja. Titik akhir kedua ini berisi nama indeks, sehingga Anda tidak perlu menentukan satu di isi permintaan.

OpenSearch Dasbor sangat bergantung pada mget dan msearch, jadi tidak mungkin berfungsi dengan baik setelah perubahan ini. Untuk remediasi parsial, Anda bisa meninggalkan `rest.action.multi.allow_explicit_index` sebagai benar dan menolak akses pengguna tertentu ke satu API atau lebih.

Untuk informasi tentang perubahan pengaturan ini, lihat the section called “Pengaturan cluster lanjutan”.

Demikian pula, kebijakan berbasis sumber daya berikut berisi dua masalah halus:

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Effect": "Allow",
      "Principal": {
        "AWS": "arn:aws:iam::123456789012:user/test-user"
      },
      "Action": "es:ESHttp*",
      "Resource": "arn:aws:es:us-west-1:987654321098:domain/test-domain/*"
    },
    {
      "Effect": "Deny",
      "Principal": {
        "AWS": "arn:aws:iam::123456789012:user/test-user"
      },
      "Action": "es:ESHttp*",
      "Resource": "arn:aws:es:us-west-1:987654321098:domain/test-domain/restricted-
index/*"
    }
  ]
```

```
}
```

- Meskipun secara eksplisit menolak, `test-user` masih dapat melakukan panggilan seperti `GET https://search-test-domain.us-west-1.es.amazonaws.com/_all/_search` dan `GET https://search-test-domain.us-west-1.es.amazonaws.com/*/_search` untuk mengakses dokumen di `restricted-index`.
- Karena `Resource` referensi elemen `restricted-index/*`, `test-user` tidak memiliki izin untuk langsung mengakses dokumen indeks. Namun, pengguna memiliki izin untuk menghapus seluruh indeks. Untuk mencegah akses dan penghapusan, kebijakan harus menentukan `restricted-index*`.

Daripada mencampur izin luas dan penolakan terfokus, pendekatan teraman adalah mengikuti prinsip hak istimewa paling rendah dan hanya memberikan izin yang diperlukan untuk melakukan tugas. Untuk informasi selengkapnya tentang mengendalikan akses ke indeks atau OpenSearch operasi individual, lihatthe section called “Kontrol akses detail”.

> **Important**
>
> Menentukan wildcard * memungkinkan akses anonim ke domain Anda. Anda tidak disarankan menggunakan wildcard Selain itu, periksa kebijakan berikut dengan cermat untuk mengonfirmasi bahwa kebijakan tersebut tidak memberikan akses luas:
>
> - Kebijakan berbasis identitas yang dilampirkan pada AWS prinsipal terkait (misalnya, peran IAM)
> - Kebijakan berbasis sumber daya yang dilampirkan pada AWS sumber daya terkait (misalnya, kunci KMS) AWS Key Management Service

Mengonfigurasi kebijakan akses

- Untuk petunjuk tentang cara membuat atau memodifikasi kebijakan berbasis sumber daya dan IP di OpenSearch Layanan, lihat. the section called “Mengonfigurasi kebijakan akses”
- Untuk petunjuk cara membuat atau memodifikasi kebijakan berbasis identitas di IAM, lihat Membuat kebijakan IAM di Panduan Pengguna IAM.

Contoh kebijakan tambahan

Meskipun Bab ini mencakup banyak contoh kebijakan, kontrol AWS akses adalah subjek kompleks yang paling baik dipahami melalui contoh. Untuk selengkapnya, lihat Contoh kebijakan berbasis identitas IAM di Panduan Pengguna IAM.

Referensi izin API OpenSearch Layanan Amazon

Saat mengatur kontrol akses, Anda menulis kebijakan izin yang dapat dilampirkan ke identitas IAM (kebijakan berbasis identitas). Untuk informasi referensi terperinci, lihat topik berikut di Referensi Otorisasi Layanan:

- Tindakan, sumber daya, dan kunci kondisi untuk OpenSearch Layanan.
- Tindakan, sumber daya, dan kunci kondisi untuk OpenSearch Ingestion.

Referensi ini berisi informasi tentang operasi API mana yang dapat digunakan dalam kebijakan IAM. Ini juga mencakup AWS sumber daya yang dapat Anda berikan izin, dan kunci kondisi yang dapat Anda sertakan untuk kontrol akses berbutir halus.

Anda menentukan tindakan di bidang `Action` kebijakan, nilai sumber daya di bidang `Resource` kebijakan, dan syarat di bidang `Condition` kebijakan. Untuk menentukan tindakan untuk OpenSearch Service, gunakan `es:` awalan yang diikuti dengan nama operasi API (misalnya,`es:CreateDomain`). Untuk menentukan tindakan untuk OpenSearch Ingestion, gunakan `osis:` awalan yang diikuti oleh operasi API (misalnya,). `osis:CreatePipeline`

AWS kebijakan terkelola untuk Amazon OpenSearch Service

Kebijakan AWS terkelola adalah kebijakan mandiri yang dibuat dan dikelola oleh AWS. AWS Kebijakan terkelola dirancang untuk memberikan izin bagi banyak kasus penggunaan umum sehingga Anda dapat mulai menetapkan izin kepada pengguna, grup, dan peran.

Perlu diingat bahwa kebijakan AWS terkelola mungkin tidak memberikan izin hak istimewa paling sedikit untuk kasus penggunaan spesifik Anda karena tersedia untuk digunakan semua pelanggan. AWS Kami menyarankan Anda untuk mengurangi izin lebih lanjut dengan menentukan kebijakan yang dikelola pelanggan yang khusus untuk kasus penggunaan Anda.

Anda tidak dapat mengubah izin yang ditentukan dalam kebijakan AWS terkelola. Jika AWS memperbarui izin yang ditentukan dalam kebijakan AWS terkelola, pemutakhiran akan memengaruhi

semua identitas utama (pengguna, grup, dan peran) yang dilampirkan kebijakan tersebut. AWS kemungkinan besar akan memperbarui kebijakan AWS terkelola saat baru Layanan AWS diluncurkan atau operasi API baru tersedia untuk layanan yang ada.

Untuk informasi selengkapnya, lihat Kebijakan yang dikelola AWS dalam Panduan Pengguna IAM.

AmazonOpenSearchServiceFullAccess

Memberikan akses penuh ke operasi API konfigurasi OpenSearch Layanan dan sumber daya untuk file Akun AWS.

Anda dapat menemukan AmazonOpenSearchServiceFullAccesskebijakan di konsol IAM.

AmazonOpenSearchServiceReadOnlyAccess

Memberikan akses hanya-baca ke semua sumber daya OpenSearch Layanan untuk file. Akun AWS

Anda dapat menemukan AmazonOpenSearchServiceReadOnlyAccesskebijakan di konsol IAM.

AmazonOpenSearchServiceRolePolicy

Anda tidak dapat melampirkan `AmazonOpenSearchServiceRolePolicy` ke entitas IAM Anda. Kebijakan ini dilampirkan ke peran terkait layanan yang memungkinkan OpenSearch Layanan mengakses sumber daya akun. Untuk informasi selengkapnya, lihat the section called “Izin”.

Anda dapat menemukan AmazonOpenSearchServiceRolePolicykebijakan di konsol IAM.

AmazonOpenSearchServiceCognitoAccess

Memberikan izin Amazon Cognito minimum yang diperlukan untuk mengaktifkan otentikasi Cognito.

Anda dapat menemukan AmazonOpenSearchServiceCognitoAccesskebijakan di konsol IAM.

AmazonOpenSearchIngestionServiceRolePolicy

Anda tidak dapat melampirkan `AmazonOpenSearchIngestionServiceRolePolicy` ke entitas IAM Anda. Kebijakan ini dilampirkan ke peran terkait layanan yang memungkinkan OpenSearch Ingestion mengaktifkan akses VPC untuk saluran saluran konsumsi, membuat tag, dan mempublikasikan metrik terkait konsumsi ke akun Anda. CloudWatch Untuk informasi selengkapnya, lihat the section called “Menggunakan peran terkait layanan”.

Anda dapat menemukan AmazonOpenSearchIngestionServiceRolePolicykebijakan di konsol IAM.

AmazonOpenSearchIngestionFullAccess

Memberikan akses penuh ke operasi dan sumber daya API OpenSearch Ingestion untuk file. Akun AWS

Anda dapat menemukan AmazonOpenSearchIngestionFullAccesskebijakan di konsol IAM.

AmazonOpenSearchIngestionReadOnlyAccess

Memberikan akses hanya-baca ke semua sumber daya OpenSearch Ingestion untuk file. Akun AWS

Anda dapat menemukan AmazonOpenSearchIngestionReadOnlyAccesskebijakan di konsol IAM.

AmazonOpenSearchServerlessServiceRolePolicy

Memberikan Amazon CloudWatch izin minimum yang diperlukan untuk mengirim data metrik OpenSearch Tanpa Server ke. CloudWatch

Anda dapat menemukan AmazonOpenSearchServerlessServiceRolePolicykebijakan di konsol IAM.

OpenSearch Pembaruan layanan untuk kebijakan AWS terkelola

Lihat detail tentang pembaruan kebijakan AWS terkelola untuk OpenSearch Layanan sejak layanan ini mulai melacak perubahan.

Perubahan	Deskripsi	Tanggal
Diperbarui `AmazonOpenSearchServiceRolePolicy` dan `AmazonElasticsearchServiceRolePolicy`	Menambahkan izin yang diperlukan untuk peran terkait layanan untuk menetapkan dan membatalkan penetapan alamat IPv6. Kebijakan Elasticsearch yang tidak digunakan lagi juga telah diperbarui untuk memastikan kompatibilitas mundur.	18 Oktober 2023

Perubahan	Deskripsi	Tanggal
Ditambahkan `AmazonOpenSearchIngestionServiceRolePolicy`	Kebijakan baru yang memungkinkan OpenSearch Ingestion mengaktifkan akses VPC untuk saluran pipa konsumsi, membuat tag, dan mempublikasikan metrik terkait konsumsi ke akun Anda. CloudWatch Untuk kebijakan JSON, lihat konsol IAM.	26 April 2023
Ditambahkan `AmazonOpenSearchIngestionFullAccess`	Kebijakan baru yang memberikan akses penuh ke operasi dan sumber daya API OpenSearch Ingestion untuk file. Akun AWS Untuk kebijakan JSON, lihat konsol IAM.	26 April 2023
Ditambahkan `AmazonOpenSearchIngestionReadOnlyAccess`	Kebijakan baru yang memberikan akses hanya-baca ke semua sumber daya OpenSearch Ingestion untuk file. Akun AWS Untuk kebijakan JSON, lihat konsol IAM.	26 April 2023

Perubahan	Deskripsi	Tanggal
Ditambahkan `AmazonOpenSearchServerlessServiceRolePolicy`	Kebijakan baru yang memberikan izin minimum yang diperlukan untuk mengirim data metrik OpenSearch Tanpa Server. Amazon CloudWatch Untuk kebijakan JSON, lihat konsol IAM.	29 November 2022
Diperbarui `AmazonOpenSearchServiceRolePolicy` dan `AmazonElasticsearchServiceRolePolicy`	Menambahkan izin yang diperlukan untuk peran terkait layanan untuk membuat titik akhir VPC yang dikelola LayananOpenSearch . Beberapa tindakan hanya dapat dilakukan ketika permintaan berisi tag`OpenSearchManaged=true` . Kebijakan Elasticsearch yang tidak digunakan lagi juga telah diperbarui untuk memastikan kompatibilitas mundur.	7 November 2022

Perubahan	Deskripsi	Tanggal
Diperbarui `AmazonOpenSearchServiceRolePolicy` dan `AmazonElasticsearchServiceRolePolicy`	Menambahkan dukungan untuk `PutMetricData` tindakan, yang diperlukan untuk mempublikasikan metrik OpenSearch klaster ke Amazon CloudWatch. Kebijakan Elasticsearch yang tidak digunakan lagi juga telah diperbarui untuk memastikan kompatibilitas mundur. Untuk kebijakan JSON, lihat konsol IAM.	12 September 2022

Perubahan	Deskripsi	Tanggal
Diperbarui `AmazonOpenSearchServiceRolePolicy` dan `AmazonElasticsearchServiceRolePolicy`	Menambahkan dukungan untuk jenis `acm` sumber daya. Kebijakan ini memberikan izin baca-saja minimum AWS Certificate Manager (ACM) yang diperlukan untuk peran terkait layanan untuk memverifikasi dan memvalidasi sumber daya ACM guna membuat dan memperbarui domain berkemampuan titik akhir kustom. Kebijakan Elasticsearch yang tidak digunakan lagi juga telah diperbarui untuk memastikan kompatibilitas mundur.	28 Juli 2022

Perubahan	Deskripsi	Tanggal
Diperbarui `AmazonOpenSearchServiceCognitoAccess` dan `AmazonESCognitoAccess`	Menambahkan dukungan untuk `UpdateUserPoolClient` tindakan, yang diperlukan untuk menyetel konfigurasi kumpulan pengguna Cognito selama peningkatan dari Elasticsearch ke. OpenSearch Izin yang dikoreksi untuk `SetIdentityPoolRoles` tindakan untuk memungkinkan akses ke semua sumber daya. Kebijakan Elasticsearch yang tidak digunakan lagi juga telah diperbarui untuk memastikan kompatibilitas mundur.	20 Desember 2021
Diperbarui `AmazonOpenSearchServiceRolePolicy`	Menambahkan dukungan untuk jenis `security-group` sumber daya. Kebijakan ini memberikan izin minimum Amazon EC2 dan Elastic Load Balancing yang diperlukan untuk peran terkait layanan guna mengaktifkan akses VPC.	9 September 2021

Perubahan	Deskripsi	Tanggal
• Ditambahkan `AmazonOpenSearchServiceFullAccess` • Usang `AmazonESFullAccess`	Kebijakan baru ini dimaksudkan untuk menggantikan kebijakan lama. Kedua kebijakan menyediakan akses penuh ke API konfigurasi OpenSearch Layanan dan semua metode HTTP untuk OpenSearch API. Kontrol akses detail dan kebijakan berbasis sumber daya masih dapat membatasi akses.	7 September 2021
• Ditambahkan `AmazonOpenSearchServiceReadOnlyAccess` • Usang `AmazonESReadOnlyAccess`	Kebijakan baru ini dimaksudkan untuk menggantikan kebijakan lama. Kedua kebijakan menyediakan akses hanya-baca ke API konfigurasi OpenSearch Layanan (`es:Describe* ,es:List*, danes:Get*`) dan tidak ada akses ke metode HTTP untuk API. OpenSearch	7 September 2021
• Ditambahkan `AmazonOpenSearchServiceCognitoAccess` • Usang `AmazonESCognitoAccess`	Kebijakan baru ini dimaksudkan untuk menggantikan kebijakan lama. Kedua kebijakan memberikan izin Amazon Cognito minimum yang diperlukan untuk mengaktifkan otentikasi Cognito.	7 September 2021

Perubahan	Deskripsi	Tanggal
• Ditambahkan AmazonOpenSearchServiceRolePolicy • Usang `AmazonElasticsearchServiceRolePolicy`	Kebijakan baru ini dimaksudkan untuk mengganti kan kebijakan lama. Kedua kebijakan tersebut memberikan izin minimum Amazon EC2 dan Elastic Load Balancing yang diperlukan untuk peran terkait layanan guna mengaktifkan akses VPC.	7 September 2021
Mulai melacak perubahan	Amazon OpenSearch Service sekarang melacak perubahan pada kebijakan AWS-managed.	7 September 2021

Cross-service bingung wakil pencegahan

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 tersebut. MasukAWS, peniruan lintas layanan dapat mengakibatkan masalah wakil bingung. Peniruan lintas layanan dapat terjadi ketika satu layanan (yanglayanan panggilan) panggilan layanan lain (yangdisebut layanan). Layanan panggilan dapat dimanipulasi untuk menggunakan izin untuk bertindak atas sumber daya pelanggan lain dengan cara yang seharusnya tidak memiliki izin untuk mengakses. Untuk mencegah hal ini,AWSmenyediakan alat yang membantu Anda melindungi data Anda untuk semua layanan dengan prinsipal layanan yang telah diberikan akses ke sumber daya di akun Anda.

Sebaiknya gunakan`aws:SourceArn`dan`aws:SourceAccount`kunci konteks kondisi global dalam kebijakan sumber daya untuk membatasi izin yang AmazonOpenSearchLayanan memberikan layanan lain untuk sumber daya. Jika`aws:SourceArn`nilai tidak mengandung ID akun, seperti bucket ARN Amazon S3, Anda harus menggunakan kedua kunci konteks kondisi global untuk membatasi izin. Jika Anda menggunakan kedua kunci konteks kondisi global dan`aws:SourceArn`nilai berisi ID akun,`aws:SourceAccount`nilai dan akun di`aws:SourceArn`nilai

harus menggunakan ID akun yang sama bila digunakan dalam pernyataan kebijakan yang sama. Gunakan`aws:SourceArn`jika Anda ingin hanya satu sumber daya yang terkait dengan akses lintas layanan. Gunakan`aws:SourceAccount`jika Anda ingin mengizinkan sumber daya apa pun di akun tersebut terkait dengan penggunaan lintas layanan.

Nilai dari`aws:SourceArn`harus menjadi ARN dariOpenSearchDomain layanan.

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

Contoh berikut menunjukkan cara menggunakan`aws:SourceArn`dan`aws:SourceAccount`kunci konteks kondisi globalOpenSearchService untuk mencegah masalah bingung deputy.

```
{
   "Version":"2012-10-17",
   "Statement":{
      "Sid":"ConfusedDeputyPreventionExamplePolicy",
      "Effect":"Allow",
      "Principal":{
         "Service":"es.amazonaws.com"
      },
      "Action":"sts:AssumeRole",
      "Condition":{
         "StringEquals":{
            "aws:SourceAccount":"123456789012"
         },
         "ArnLike":{
            "aws:SourceArn":"arn:aws:es:region:123456789012:domain/my-domain"
         }
      }
   }
}
```

Kontrol akses berbutir halus di Layanan Amazon OpenSearch

Kontrol akses berbutir halus menawarkan cara tambahan untuk mengontrol akses ke data Anda di Layanan Amazon. OpenSearch Misalnya, bergantung pada siapa yang membuat permintaan,

Anda mungkin ingin penelusuran mengembalikan hasil hanya dari satu indeks. Anda mungkin ingin menyembunyikan bidang tertentu dalam dokumen Anda atau mengecualikan dokumen tertentu sama sekali.

Kontrol akses detail menawarkan manfaat sebagai berikut:

- Kontrol akses berbasis peran
- Keamanan pada tingkat indeks, dokumen, dan bidang
- OpenSearch Dasbor multi-tenancy
- Otentikasi dasar HTTP untuk OpenSearch dan OpenSearch Dasbor

Topik

Gambaran yang lebih besar: kontrol akses berbutir halus dan keamanan Layanan OpenSearch

Keamanan Amazon OpenSearch Service memiliki tiga lapisan utama:

Jaringan

Lapisan keamanan pertama adalah jaringan, yang menentukan apakah permintaan mencapai domain OpenSearch Layanan. Jika Anda memilih Akses publik ketika Anda membuat domain, permintaan dari setiap klien yang terhubung internet dapat mencapai titik akhir domain. Jika Anda memilih Akses VPC, klien harus terhubung ke VPC (dan kelompok keamanan terkait harus mengizinkannya) untuk permintaan untuk mencapai titik akhir. Untuk informasi selengkapnya, lihat the section called “Dukungan VPC”.

Kebijakan akses domain

Lapisan keamanan kedua adalah kebijakan akses domain. Setelah permintaan mencapai titik akhir domain, kebijakan akses berbasis sumber daya memungkinkan atau menyangkal akses permintaan ke URI yang diberikan. Kebijakan akses menerima atau menolak permintaan di “tepi” domain, sebelum mereka mencapai OpenSearch dirinya sendiri.

Kontrol akses berbutir halus

Lapisan keamanan ketiga dan terakhir adalah kontrol akses yang sangat baik. Setelah kebijakan akses berbasis sumber daya memungkinkan permintaan untuk mencapai titik akhir domain, kontrol akses detail mengevaluasi kredensial pengguna dan mengautentikasi pengguna atau menolak permintaan. Jika kontrol akses detail mengautentikasi pengguna, mengambil semua peran yang dipetakan ke pengguna itu dan menggunakan set lengkap izin untuk menentukan bagaimana menangani permintaan.

> **Note**
>
> Jika kebijakan akses berbasis sumber daya berisi peran atau pengguna IAM, klien harus mengirim permintaan yang ditandatangani menggunakan Tanda Tangan Versi 4. AWS Dengan demikian, kebijakan akses dapat bertentangan dengan kontrol akses detail, terutama jika Anda menggunakan basis data pengguna internal dan autentikasi basic HTTP. Anda tidak dapat menandatangani permintaan dengan nama pengguna dan kata sandi serta kredensil IAM. Secara umum, jika Anda mengaktifkan kontrol akses detail, sebaiknya gunakan kebijakan akses domain yang tidak memerlukan permintaan yang ditandatangani.

Diagram berikut mengilustrasikan konfigurasi umum: domain akses VPC dengan kontrol akses berbutir halus diaktifkan, kebijakan akses berbasis IAM, dan pengguna master IAM.

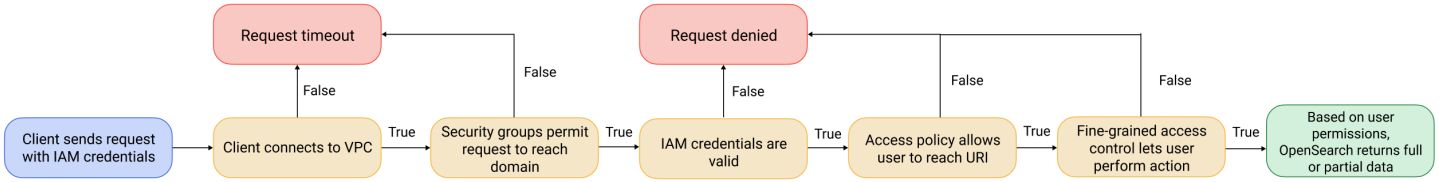
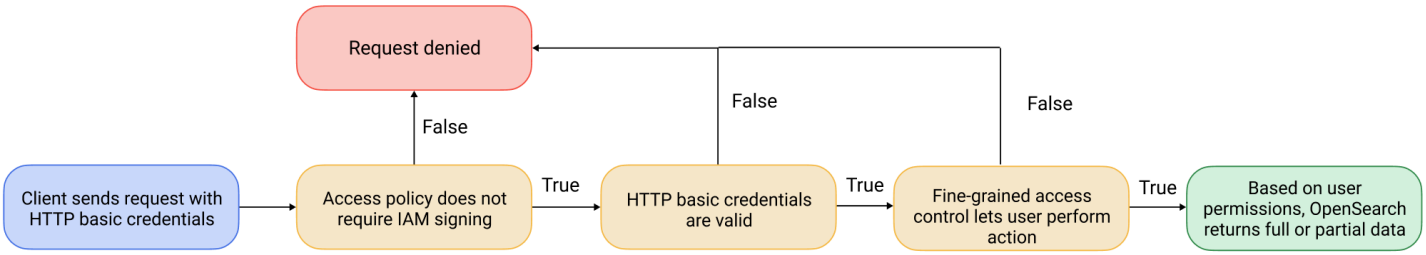


Diagram berikut menggambarkan konfigurasi umum lainnya: domain akses publik dengan kontrol akses halus diaktifkan, kebijakan akses yang tidak menggunakan prinsip IAM, dan pengguna master dalam database pengguna internal.



Contoh

Pertimbangkan permintaan `GET` ke `movies/_search?q=thor`. Apakah pengguna memiliki izin untuk mencari indeks `movies`? Jika demikian, apakah pengguna memiliki izin untuk melihat semua dokumen di dalamnya? Haruskah respons menghilangkan atau menganonimkan bidang apa pun? Untuk pengguna utama, responsnya mungkin akan terlihat seperti ini:

```
{
  "hits": {
    "total": 7,
    "max_score": 8.772789,
    "hits": [{
        "_index": "movies",
        "_type": "_doc",
        "_id": "tt0800369",
        "_score": 8.772789,
        "_source": {
          "directors": [
            "Kenneth Branagh",
            "Joss Whedon"
          ],
          "release_date": "2011-04-21T00:00:00Z",
          "genres": [
```

```
            "Action",
            "Adventure",
            "Fantasy"
          ],
          "plot": "The powerful but arrogant god Thor is cast out of Asgard to
 live amongst humans in Midgard (Earth), where he soon becomes one of their finest
 defenders.",
          "title": "Thor",
          "actors": [
            "Chris Hemsworth",
            "Anthony Hopkins",
            "Natalie Portman"
          ],
          "year": 2011
        }
      },
      ...
    ]
  }
}
```

Jika pengguna dengan izin yang lebih terbatas mengeluarkan permintaan yang sama persis, responsnya mungkin terlihat seperti ini:

```
{
  "hits": {
    "total": 2,
    "max_score": 8.772789,
    "hits": [{
        "_index": "movies",
        "_type": "_doc",
        "_id": "tt0800369",
        "_score": 8.772789,
        "_source": {
          "year": 2011,
          "release_date":
 "3812a72c6dd23eef3c750c2d99e205cbd260389461e19d610406847397ecb357",
          "plot": "The powerful but arrogant god Thor is cast out of Asgard to
 live amongst humans in Midgard (Earth), where he soon becomes one of their finest
 defenders.",
          "title": "Thor"
        }
      },
```

```
      ...
    ]
  }
}
```

Respons memiliki lebih sedikit klik dan lebih sedikit bidang untuk setiap klik. Juga, bidang `release_date` dianonimkan. Jika pengguna tanpa izin membuat permintaan yang sama, klaster mengembalikan kesalahan:

```
{
  "error": {
    "root_cause": [{
      "type": "security_exception",
      "reason": "no permissions for [indices:data/read/search] and User [name=limited-
user, roles=[], requestedTenant=null]"
    }],
    "type": "security_exception",
    "reason": "no permissions for [indices:data/read/search] and User [name=limited-
user, roles=[], requestedTenant=null]"
  },
  "status": 403
}
```

Jika pengguna memberikan kredensial yang tidak valid, klaster mengembalikan pengecualian `Unauthorized`.

Konsep utama

Saat Anda memulai dengan kontrol akses berbutir halus, pertimbangkan konsep-konsep berikut:

- Peran — Cara inti menggunakan kontrol akses berbutir halus. Dalam hal ini, peran berbeda dari peran IAM. Peran berisi kombinasi izin: klaster-lebar, indeks spesifik, tingkat dokumen, dan tingkat bidang.
- Pemetaan — Setelah mengonfigurasi peran, Anda memetakannya ke satu atau beberapa pengguna. Misalnya, Anda dapat memetakan tiga peran ke satu pengguna: satu peran yang menyediakan akses ke Dasbor, satu yang menyediakan akses hanya-baca`index1`, dan satu yang menyediakan akses tulis. `index2` Atau Anda dapat menyertakan semua izin tersebut dalam satu peran.

- Pengguna — Orang atau aplikasi yang membuat permintaan ke OpenSearch cluster. Pengguna memiliki kredensial—baik kunci akses IAM atau nama pengguna dan kata sandi—yang mereka tentukan saat mereka membuat permintaan.

Tentang pengguna master

Pengguna utama di OpenSearch Layanan adalah kombinasi nama pengguna dan kata sandi, atau prinsipal IAM, yang memiliki izin penuh ke cluster yang mendasarinya OpenSearch. Seorang pengguna dianggap sebagai pengguna utama jika mereka memiliki semua akses ke OpenSearch cluster bersama dengan kemampuan untuk membuat pengguna internal, peran, dan pemetaan peran dalam OpenSearch Dasbor.

Pengguna master yang dibuat di konsol OpenSearch Layanan atau melalui CLI secara otomatis dipetakan ke dua peran yang telah ditentukan:

- `all_access`— Menyediakan akses penuh ke semua operasi di seluruh cluster, izin untuk menulis ke semua indeks cluster, dan izin untuk menulis ke semua penyewa.
- `security_manager`— Menyediakan akses ke plugin Keamanan dan manajemen pengguna dan izin.

Dengan dua peran ini, pengguna mendapatkan akses ke tab Keamanan di OpenSearch Dasbor, tempat mereka dapat mengelola pengguna dan izin. Jika Anda membuat pengguna internal lain dan hanya memetakannya ke `all_access` peran, pengguna tidak memiliki akses ke tab Keamanan. Anda dapat membuat pengguna master tambahan dengan memetakannya secara eksplisit ke peran dan peran. `all_access` `security_manager` Untuk petunjuk, lihat the section called “Pengguna utama tambahan”.

Saat Anda membuat pengguna master untuk domain Anda, Anda dapat menentukan salah satu prinsipal IAM yang ada, atau membuat pengguna master dalam database pengguna internal. Pertimbangkan hal berikut saat memutuskan mana yang akan digunakan:

- Prinsipal IAM — Jika Anda memilih prinsipal IAM untuk pengguna master Anda, semua permintaan ke klaster harus ditandatangani menggunakan AWS Signature Version 4.

  OpenSearch Layanan tidak mempertimbangkan izin kepala sekolah IAM apa pun. Pengguna atau peran IAM berfungsi murni untuk otentikasi. Kebijakan tentang pengguna atau peran tersebut tidak ada kaitannya dengan otorisasi pengguna utama. Otorisasi ditangani melalui berbagai izin di plugin Keamanan. OpenSearch

Misalnya, Anda dapat menetapkan nol izin IAM ke prinsipal IAM, dan selama mesin atau orang tersebut dapat mengautentikasi ke pengguna atau peran tersebut, mereka memiliki kekuatan pengguna utama di Layanan. OpenSearch

Kami merekomendasikan IAM jika Anda ingin menggunakan pengguna yang sama di beberapa cluster, jika Anda ingin menggunakan Amazon Cognito untuk mengakses Dasbor, atau jika Anda OpenSearch memiliki klien yang mendukung penandatanganan Signature Version 4.

- Database pengguna internal — Jika Anda membuat master di database pengguna internal (dengan kombinasi nama pengguna dan kata sandi), Anda dapat menggunakan otentikasi dasar HTTP (serta kredenal IAM) untuk membuat permintaan ke cluster. Sebagian besar klien mendukung otentikasi dasar, termasuk curl, yang juga mendukung AWS Signature Version 4 dengan opsi --aws-sigv4. Database pengguna internal disimpan dalam OpenSearch indeks, sehingga Anda tidak dapat membagikannya dengan cluster lain.

  Kami merekomendasikan database pengguna internal jika Anda tidak perlu menggunakan kembali pengguna di beberapa cluster, jika Anda ingin menggunakan otentikasi dasar HTTP untuk mengakses Dasbor (bukan Amazon Cognito), atau jika Anda memiliki klien yang hanya mendukung otentikasi dasar. Database pengguna internal adalah cara paling sederhana untuk memulai dengan OpenSearch Layanan.

Mengaktifkan kontrol akses detail

Aktifkan kontrol akses berbutir halus menggunakan konsol, AWS CLI, atau API konfigurasi. Untuk langkah, lihat Membuat dan mengelola domain.

Kontrol akses berbutir halus memerlukan OpenSearch atau Elasticsearch 6.7 atau yang lebih baru. Ini juga membutuhkan HTTPS untuk semua lalu lintas ke domain, Enkripsi data saat istirahat, dan node-to-node enkripsi. Bergantung pada cara Anda mengonfigurasi fitur lanjutan dari kontrol akses berbutir halus, pemrosesan tambahan permintaan Anda mungkin memerlukan sumber daya komputasi dan memori pada node data individual. Setelah Anda mengaktifkan kontrol akses detail, Anda tidak dapat menonaktifkannya.

Mengaktifkan kontrol akses berbutir halus pada domain yang ada

Anda dapat mengaktifkan kontrol akses berbutir halus pada domain yang ada yang berjalan OpenSearch atau Elasticsearch 6.7 atau yang lebih baru.

Untuk mengaktifkan kontrol akses berbutir halus pada domain yang ada (konsol)

1. Pilih domain Anda dan pilih Tindakan dan Edit konfigurasi keamanan.
2. Pilih Aktifkan kontrol akses berbutir halus.
3. Pilih cara membuat pengguna master:
   - Jika Anda ingin menggunakan IAM untuk manajemen pengguna, pilih Setel IAM ARN sebagai pengguna utama dan tentukan ARN untuk peran IAM.
   - Jika Anda ingin menggunakan database pengguna internal, pilih Buat pengguna utama dan tentukan nama pengguna dan kata sandi.
4. (Opsional) Pilih Aktifkan periode migrasi untuk kebijakan akses berbasis Buka/IP. Pengaturan ini memungkinkan periode transisi 30 hari di mana pengguna Anda yang ada dapat terus mengakses domain tanpa gangguan, dan kebijakan akses terbuka dan berbasis IP yang ada akan terus bekerja dengan domain Anda. Selama periode migrasi ini, kami menyarankan agar administrator membuat peran yang diperlukan dan memetakannya ke pengguna untuk domain. Jika Anda menggunakan kebijakan berbasis identitas alih-alih kebijakan akses terbuka atau berbasis IP, Anda dapat menonaktifkan setelan ini.

   Anda juga perlu memperbarui klien Anda untuk bekerja dengan kontrol akses berbutir halus selama periode migrasi. Misalnya, jika Anda memetakan peran IAM dengan kontrol akses berbutir halus, Anda harus memperbarui klien Anda untuk mulai menandatangani permintaan dengan AWS Signature Version 4. Jika Anda mengonfigurasi otentikasi dasar HTTP dengan kontrol akses berbutir halus, Anda harus memperbarui klien Anda untuk memberikan kredensyal otentikasi dasar yang sesuai dalam permintaan.

   Selama periode migrasi, pengguna yang mengakses titik akhir OpenSearch Dasbor untuk domain akan langsung mendarat di halaman Discover, bukan halaman login. Administrator dan pengguna master dapat memilih Login untuk masuk dengan kredensi admin dan mengonfigurasi pemetaan peran.

   > **Important**
   >
   > OpenSearch Layanan secara otomatis menonaktifkan periode migrasi setelah 30 hari. Kami menyarankan untuk mengakhirinya segera setelah Anda membuat peran yang diperlukan dan memetakannya kepada pengguna. Setelah periode migrasi berakhir, Anda tidak dapat mengaktifkannya kembali.

5. Pilih Simpan perubahan.

Perubahan tersebut memicu penerapan biru/hijau di mana kesehatan cluster menjadi merah, tetapi semua operasi cluster tetap tidak terpengaruh.

Untuk mengaktifkan kontrol akses berbutir halus pada domain yang ada (CLI)

Setel `AnonymousAuthEnabled` `true` untuk mengaktifkan periode migrasi dengan kontrol akses berbutir halus:

```
aws opensearch update-domain-config --domain-name test-domain --region us-east-1 \
      --advanced-security-options '{ "Enabled": true,
 "InternalUserDatabaseEnabled":true, "MasterUserOptions": {"MasterUserName":"master-
username","MasterUserPassword":"master-password"},"AnonymousAuthEnabled": true}'
```

Tentang default_role

Kontrol akses berbutir halus membutuhkan pemetaan peran. Jika domain Anda menggunakan kebijakan akses berbasis identitas, OpenSearch Layanan secara otomatis memetakan pengguna Anda ke peran baru yang disebut default_role untuk membantu Anda memigrasi pengguna yang sudah ada dengan benar. Pemetaan sementara ini memastikan bahwa pengguna Anda masih dapat berhasil mengirim permintaan GET dan PUT yang ditandatangani oleh IAM hingga Anda membuat pemetaan peran Anda sendiri.

Peran tersebut tidak menambahkan kerentanan atau kekurangan keamanan apa pun ke domain OpenSearch Layanan Anda. Sebaiknya hapus peran default segera setelah Anda mengatur peran Anda sendiri dan memetakannya sesuai dengan itu.

Skenario migrasi

Tabel berikut menjelaskan perilaku untuk setiap metode otentikasi sebelum dan sesudah mengaktifkan kontrol akses berbutir halus pada domain yang ada, dan langkah-langkah yang harus diambil administrator untuk memetakan pengguna mereka dengan benar ke peran:

Metode otentikasi	Sebelum mengaktifkan kontrol akses berbutir halus	Setelah mengaktifkan kontrol akses berbutir halus	Tugas administrator
Kebijakan berbasis identitas	Semua pengguna yang memenuhi kebijakan IAM dapat mengakses domain.	Anda tidak perlu mengaktifkan periode migrasi. OpenSearch Layanan secara otomatis memetakan semua pengguna yang memenuhi kebijakan IAM ke default_role sehingga mereka dapat terus mengakses domain.	1. Buat pemetaan peran khusus di domain. 2. Hapus default_role.
Kebijakan berbasis IP	Semua pengguna dari alamat IP yang diizinkan atau blok CIDR dapat mengakses domain.	Selama periode migrasi 30 hari, semua pengguna dari alamat IP yang diizinkan atau blok CIDR dapat terus mengakses domain.	1. Buat pemetaan peran khusus di domain. 2. Perbarui klien Anda untuk menyediakan kredensyal otentikasi dasar atau kredensyal IAM, tergantung pada konfigurasi pemetaan peran Anda. 3. Nonaktifkan periode migrasi. Pengguna dari alamat IP yang diizinkan atau blok CIDR mengirim permintaan tanpa otentikasi dasar atau kredensyal IAM akan kehilangan akses ke domain.
Kebijakan akses terbuka	Semua pengguna melalui internet dapat	Selama periode migrasi 30 hari, semua pengguna melalui internet dapat	1. Buat pemetaan peran di domain. 2. Perbarui klien Anda untuk menyediakan kredensyal otentikas

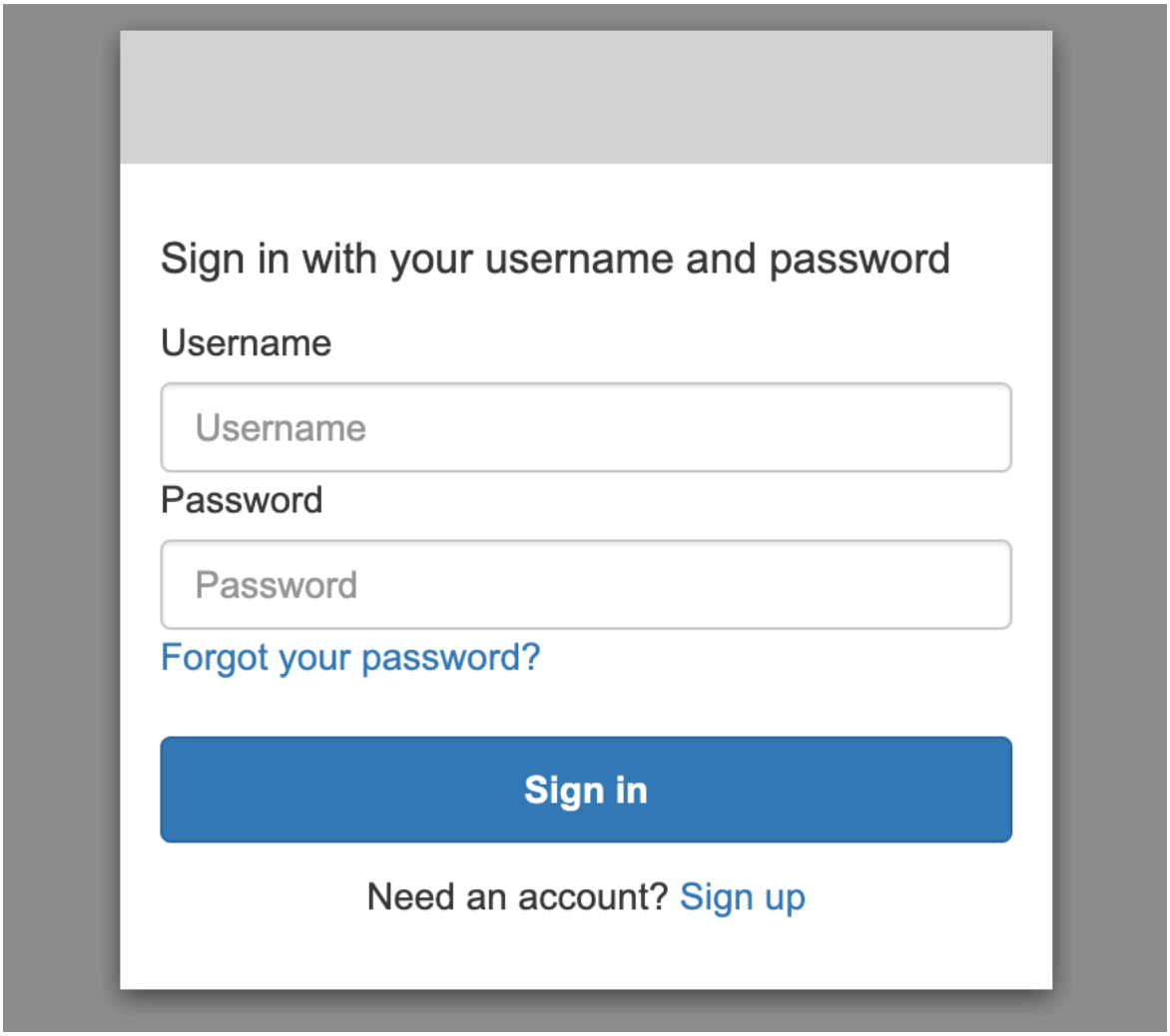
Metode otentikasi	Sebelum mengaktifkan kontrol akses berbutir halus	Setelah mengaktifkan kontrol akses berbutir halus	Tugas administrator
	mengakses domain.	terus mengakses domain.	i dasar atau kredensyal IAM, tergantung pada konfigurasi pemetaan peran Anda. 3. Nonaktifkan periode migrasi. Pengguna yang mengirim permintaan tanpa otentikasi dasar atau kredensil IAM akan kehilangan akses ke domain.

Mengakses OpenSearch Dasbor sebagai pengguna utama

Kontrol akses berbutir halus memiliki plugin OpenSearch Dasbor yang menyederhanakan tugas manajemen. Anda dapat menggunakan Dasbor untuk mengelola pengguna, peran, pemetaan, grup tindakan, dan penyewa. Namun, halaman login OpenSearch Dasbor dan metode otentikasi yang mendasarinya berbeda, tergantung pada cara Anda mengelola pengguna dan mengonfigurasi domain Anda.

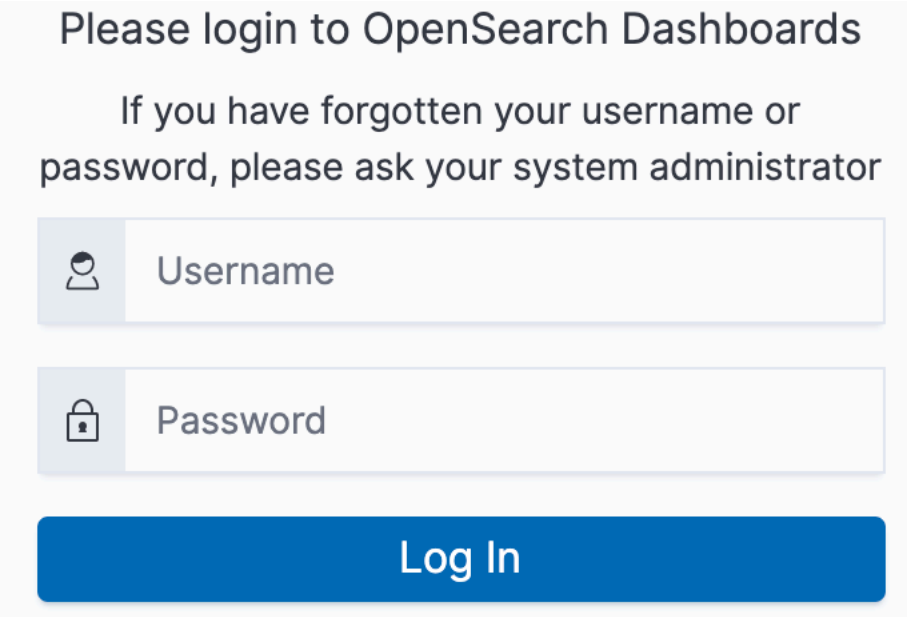
- Jika Anda ingin menggunakan IAM untuk manajemen pengguna, gunakan the section called “Otentikasi Amazon Cognito untuk Dasbor OpenSearch ” untuk mengakses Dasbor. Jika tidak, Dasbor menampilkan halaman masuk yang tidak berfungsi. Lihat the section called “Batasan”.

  Dengan autentikasi Amazon Cognito, salah satu peran diasumsikan dari kolam identitas harus sesuai dengan IAM role yang Anda tentukan untuk pengguna master. Untuk informasi lebih lanjut tentang konfigurasi ini, lihat the section called “(Opsional) Mengonfigurasi akses terperinci” dan the section called “Tutorial: Kontrol akses berbutir halus dengan otentikasi Cognito”.



- Jika Anda memilih untuk menggunakan basis data pengguna internal, Anda dapat masuk ke Dasbor dengan nama pengguna dan kata sandi utama Anda. Anda harus mengakses Dasbor melalui HTTPS. Otentikasi Amazon Cognito dan SAM untuk Dasbor keduanya menggantikan layar login ini.

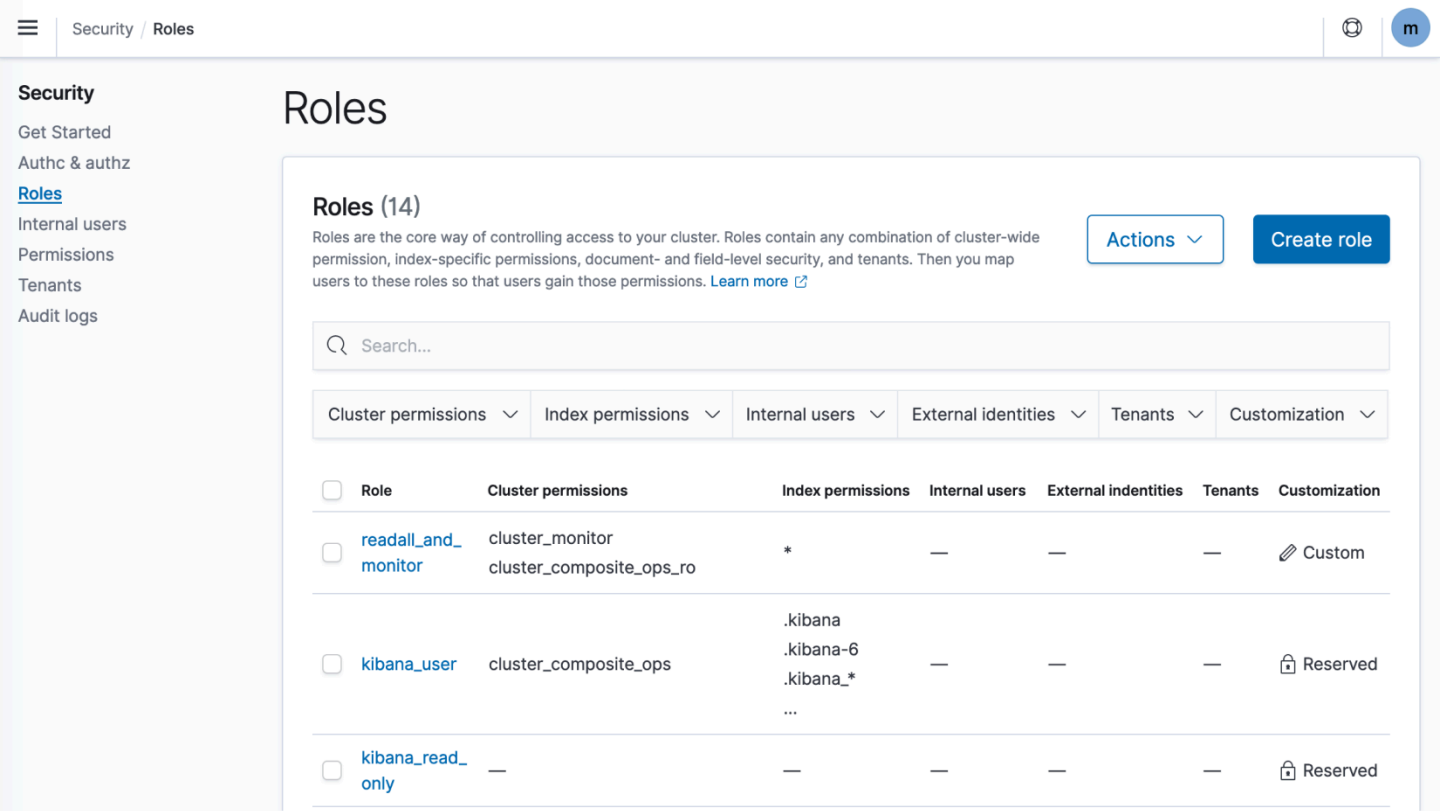
  Untuk informasi lebih lanjut tentang konfigurasi ini, lihat the section called “Tutorial: Database pengguna internal dengan otentikasi dasar”.



- Jika Anda memilih untuk menggunakan autentikasi SAML, Anda dapat masuk menggunakan kredensial dari penyedia identitas eksternal. Untuk informasi selengkapnya, lihat the section called “Otentikasi SAMP untuk Dasbor OpenSearch ”.

Mengelola izin

Sebagaimana dicatat di the section called “Konsep utama”, Anda mengelola izin kontrol akses detail menggunakan peran, pengguna, dan pemetaan. Bagian ini menjelaskan cara membuat dan menerapkan sumber daya tersebut. Kami menyarankan Anda masuk ke Dasbor sebagai pengguna utama untuk melakukan operasi ini.



> **Note**
>
> Izin yang Anda pilih untuk diberikan kepada pengguna Anda sangat bervariasi berdasarkan kasus penggunaan. Kami tidak dapat secara layak mencakup semua skenario dalam dokumentasi ini. Saat Anda menentukan izin mana yang akan diberikan kepada pengguna Anda, pastikan untuk mereferensikan izin OpenSearch klaster dan indeks yang disebutkan di bagian berikut, dan selalu ikuti prinsip hak istimewa paling sedikit.

Membuat peran

Anda dapat membuat peran baru untuk kontrol akses berbutir halus menggunakan OpenSearch Dasbor atau `_plugins/_security` operasi di REST API. Untuk informasi selengkapnya, lihat Membuat peran.

Kontrol akses detail juga mencakup sejumlah peran yang telah ditetapkan. Klien seperti OpenSearch Dasbor dan Logstash membuat berbagai macam permintaan OpenSearch, yang dapat menyulitkan untuk membuat peran secara manual dengan set izin minimum. Misalnya, peran `opensearch_dashboards_user` menyertakan izin yang pengguna perlu bekerja dengan pola

indeks, visualisasi, dashboard, dan penyewa. Kami merekomendasikan untuk memetakannya ke setiap pengguna atau peran backend yang mengakses Dasbor, bersama dengan peran tambahan yang memungkinkan akses ke indeks lain.

Amazon OpenSearch Service tidak menawarkan OpenSearch peran berikut:

- `observability_full_access`
- `observability_read_access`
- `reports_read_access`
- `reports_full_access`

Amazon OpenSearch Service menawarkan beberapa peran yang tidak tersedia dengan OpenSearch:

- `ultrawarm_manager`
- `ml_full_access`
- `cold_manager`
- `notifications_full_access`
- `notifications_read_access`

Keamanan tingkat klaster

Izin tingkat klaster termasuk kemampuan untuk membuat permintaan yang luas seperti `_mget`, `_msearch`, dan `_bulk`, memantau kesehatan, mengambil snapshot, dan banyak lagi. Kelola izin ini menggunakan bagian Izin cluster saat membuat peran. Untuk daftar lengkap izin tingkat klaster, lihat Izin klaster.

Alih-alih izin individual, Anda sering dapat mencapai postur keamanan yang Anda inginkan menggunakan kombinasi grup tindakan default. Untuk daftar grup aksi tingkat klaster, lihat Tingkat klaster.

Keamanan tingkat indeks

Izin tingkat indeks termasuk kemampuan untuk membuat indeks baru, indeks pencarian, membaca dan menulis dokumen, menghapus dokumen, mengelola alias, dan banyak lagi. Kelola izin ini menggunakan bagian Izin Indeks saat membuat peran. Untuk daftar lengkap izin tingkat indeks, lihat Izin indeks.

Alih-alih izin individual, Anda sering dapat mencapai postur keamanan yang Anda inginkan menggunakan kombinasi grup tindakan default. Untuk daftar grup tindakan tingkat indeks, lihat Tingkat indeks.

Keamanan tingkat dokumen

Keamanan tingkat dokumen memungkinkan Anda membatasi dokumen mana dalam indeks yang dapat dilihat pengguna. Saat membuat peran, tentukan pola indeks dan OpenSearch kueri. Setiap pengguna yang Anda petakan ke peran tersebut dapat melihat hanya dokumen yang cocok dengan kueri. Keamanan tingkat dokumen mempengaruhi jumlah klik yang Anda terima saat Anda mencari.

Untuk informasi selengkapnya, lihat Keamanan tingkat dokumen.

Keamanan tingkat bidang

Keamanan tingkat bidang memungkinkan Anda mengontrol bidang dokumen mana yang dapat dilihat pengguna. Saat membuat peran, tambahkan daftar bidang untuk disertakan atau dikecualikan. Jika Anda menyertakan bidang, setiap pengguna yang Anda petakan ke peran tersebut hanya dapat melihat bidang tersebut. Jika Anda mengecualikan bidang, mereka dapat melihat semua bidang kecuali yang dikecualikan. Keamanan tingkat bidang memengaruhi jumlah bidang yang disertakan dalam klik saat Anda mencari..

Untuk informasi selengkapnya, lihat Keamanan tingkat lapangan.

Penyamaran bidang

Penyamaran bidang adalah alternatif untuk keamanan tingkat bidang yang memungkinkan Anda menganonimkan data di bidang daripada menghapusnya sama sekali. Saat membuat peran, tambahkan daftar bidang untuk disamarkan. Kolom penyamaran memengaruhi apakah Anda dapat melihat isi dari suatu bidang saat mencari.

> **Tip**
>
> Jika Anda menerapkan masking standar ke bidang, OpenSearch Layanan menggunakan hash acak yang aman yang dapat menyebabkan hasil agregasi yang tidak akurat. Untuk melakukan agregasi pada bidang yang disamarkan, gunakan penyamaran berbasis pola sebagai gantinya.

Membuat pengguna

Jika Anda mengaktifkan database pengguna internal, Anda dapat membuat pengguna menggunakan OpenSearch Dasbor atau `_plugins/_security` operasi di REST API. Untuk informasi selengkapnya, lihat Membuat pengguna.

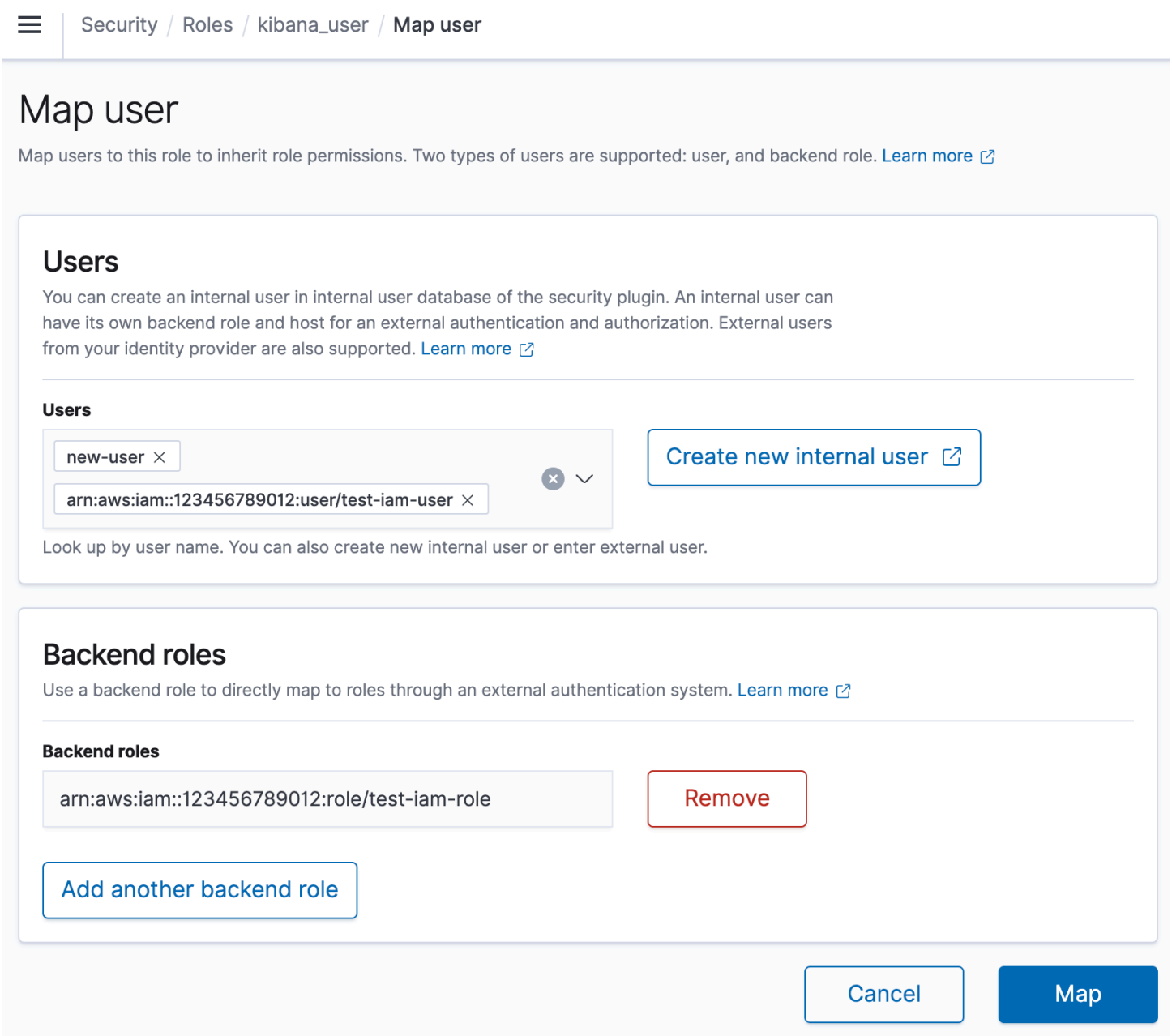
Jika Anda memilih IAM untuk pengguna master Anda, abaikan bagian Dasbor ini. Buat peran IAM sebagai gantinya. Untuk informasi lebih lanjut, lihat Panduan Pengguna IAM.

Memetakan peran untuk pengguna

Pemetaan peran adalah aspek yang paling penting dari kontrol akses detail. Kontrol akses detail memiliki beberapa peran yang telah ditetapkan untuk membantu Anda memulai, tetapi kecuali jika Anda memetakan peran untuk pengguna, setiap permintaan ke klaster berakhir dengan kesalahan izin.

Peran backend dapat membantu menyederhanakan proses pemetaan peran. Daripada memetakan peran yang sama ke 100 pengguna individu, Anda dapat memetakan peran tersebut ke peran backend tunggal yang dibagikan oleh 100 pengguna. Peran backend dapat berupa peran IAM atau string arbitrer.

- Tentukan pengguna, ARN pengguna, dan string pengguna Amazon Cognito di bagian Pengguna. String pengguna Cognito berbentuk `Cognito/`*`user-pool-id`*`/`*`username`*.
- Tentukan peran backend dan ARN peran IAM di bagian peran Backend.



Anda dapat memetakan peran ke pengguna menggunakan OpenSearch Dasbor atau `_plugins/_security` operasi di REST API. Untuk informasi selengkapnya, lihat Memetakan pengguna ke peran.

Membuat grup tindakan

Grup tindakan adalah kumpulan izin yang dapat Anda gunakan kembali di sumber daya yang berbeda. Anda dapat membuat grup tindakan baru menggunakan OpenSearch Dasbor atau `_plugins/_security` operasi di REST API, meskipun grup tindakan default cukup untuk sebagian

besar kasus penggunaan. Untuk informasi selengkapnya tentang grup tindakan default, lihat Grup tindakan default.

OpenSearch Dasbor multi-tenancy

Penyewa adalah ruang untuk menyimpan pola indeks, visualisasi, dasbor, dan objek Dasbor lainnya. Dashboard multi-tenancy memungkinkan Anda berbagi pekerjaan dengan aman dengan pengguna Dasbor lain (atau menjaganya tetap pribadi) dan mengonfigurasi penyewa secara dinamis. Anda dapat mengontrol peran yang memiliki akses ke penyewa dan apakah peran tersebut telah membaca atau menulis akses. Penyewa global adalah default. Untuk mempelajari lebih lanjut, lihat OpenSearch Dasbor multi-tenancy.

Untuk melihat penyewa Anda saat ini atau mengubah penyewa

1. Arahkan ke OpenSearch Dasbor dan masuk.
2. Pilih ikon pengguna Anda di kanan atas dan pilih Beralih penyewa.
3. Verifikasi penyewa Anda sebelum membuat visualisasi atau dashboard. Jika Anda ingin berbagi pekerjaan Anda dengan semua pengguna Dasbor lainnya, pilih Global. Untuk membagikan pekerjaan Anda dengan subset pengguna Dasbor, pilih penyewa bersama yang berbeda. Jika tidak, pilih Privat.

> **Note**
>
> OpenSearch Dasbor mempertahankan indeks terpisah untuk setiap penyewa, dan membuat template indeks yang disebut. `tenant_template` Jangan menghapus atau memodifikasi `tenant_template` indeks, karena dapat menyebabkan OpenSearch Dasbor tidak berfungsi jika pemetaan indeks penyewa salah dikonfigurasi.

Konfigurasi yang direkomendasikan

Karena kontrol akses detail berinteraksi dengan fitur keamanan lainnya, kami merekomendasikan beberapa konfigurasi kontrol akses detail yang bekerja dengan baik untuk sebagian besar kasus penggunaan.

Deskripsi	Pengguna utama	Kebijakan akses domain

Gunakan kredensyal IAM untuk panggilan ke OpenSearch API, dan gunakan otentikasi SAFL untuk mengakses Dasbor. Kelola peran kontrol akses berbutir halus menggunakan Dasbor atau REST API.	Peran IAM atau pengguna	<pre>{
		  "Version": "2012-10-17",
		  "Statement": [
		  {
		  "Effect": "Allow",
		  "Principal": {
		  "AWS": "*"
		  },
		  "Action": "es:ESHttp*",
		  "Resource": " *domain-arn* /*"
		  }
		  ]
		}</pre>

Gunakan kredenal IAM atau otentikasi dasar untuk panggilan ke API. OpenSearch Kelola peran kontrol akses berbutir halus menggunakan Dasbor atau REST API.
 	
Konfigurasi ini menawarkan banyak fleksibilitas, terutama jika Anda memiliki OpenSearch klien yang hanya mendukung otentikasi dasar.
 	
Jika Anda memiliki penyedia identitas yang sudah ada, gunakan otentikasi SAM untuk mengakses Dasbor. Jika
	Nama pengguna dan kata sandi	<pre>{
		  "Version": "2012-10-17",
		  "Statement": [
		  {
		  "Effect": "Allow",
		  "Principal": {
		  "AWS": "*"
		  },
		  "Action": "es:ESHttp*",
		  "Resource": " *domain-arn* /*"
		  }
		  ]
		}</pre>

Deskripsi	Pengguna utama	Kebijakan akses domain
tidak, kelola pengguna Dasbor di database pengguna internal.		
Gunakan kredenal IAM untuk panggilan ke OpenSearch API, dan gunakan Amazon Cognito untuk mengakses Dasbor. Kelola peran kontrol akses berbutir halus menggunakan Dasbor atau REST API.	Peran IAM atau pengguna	`{ "Version": "2012-10-17", "Statement": [ { "Effect": "Allow", "Principal": { "AWS": "*" }, "Action": "es:ESHttp*", "Resource": " domain-arn /*" } ] }`

Deskripsi	Pengguna utama	Kebijakan akses domain

Gunakan kredenal IAM untuk panggilan ke OpenSearch API, dan blokir sebagian besar akses ke Dasbor. Mengelola peran kontrol akses detail menggunak an API REST.	Peran IAM atau pengguna	{
		"Version": "2012-10-17",
		"Statement": [
		{
		"Effect": "Allow",
		"Principal": {
		"AWS": "*"
		},
		"Action": "es:ESHttp*",
		"Resource": " *domain-arn* /*"
		},
		{
		"Effect": "Deny",
		"Principal": {
		"AWS": "*"
		},
		"Action": "es:ESHttp*",
		"Resource": " *domain-arn* /_dashboards*"
		}
		]
		}

Batasan

Kontrol akses detail memiliki beberapa keterbatasan penting:

- Aspek `hosts` dari pemetaan peran, yang memetakan peran ke nama host atau alamat IP, tidak berfungsi jika domain berada dalam VPC. Anda masih dapat memetakan peran untuk pengguna dan peran backend.
- Jika Anda memilih IAM untuk pengguna master dan tidak mengaktifkan autentikasi Amazon Cognito atau SAFL, Dasbor akan menampilkan halaman login yang tidak berfungsi.
- Jika Anda memilih IAM untuk pengguna utama, Anda masih dapat membuat pengguna dalam basis data pengguna internal. Karena autentikasi basic HTTP tidak diaktifkan pada konfigurasi ini, namun, setiap permintaan ditandatangani dengan kredensial pengguna tersebut ditolak.

- Jika Anda menggunakan SQL untuk kueri indeks yang Anda tidak memiliki aksesnya, Anda menerima kesalahn "tidak ada izin". Jika indeks tidak ada, Anda menerima kesalahan “indeks tersebut tidak ada”. Perbedaan pesan kesalahan ini berarti Anda dapat mengonfirmasi keberadaan indeks jika Anda menebak namanya.

  Untuk meminimalkan masalah ini, jangan sertakan informasi sensitif dalam nama indeks. Untuk menolak semua akses ke SQL, tambahkan elemen berikut ke kebijakan akses domain Anda:

```
{
  "Effect": "Deny",
  "Principal": {
    "AWS": [
      "*"
    ]
  },
  "Action": [
    "es:*"
  ],
  "Resource": "arn:aws:es:us-east-1:123456789012:domain/my-domain/_plugins/_sql"
}
```

- Jika versi domain Anda 2.3 atau lebih tinggi dan Anda mengaktifkan kontrol akses berbutir halus, pengaturan `max_clause_count` ke 1 menyebabkan masalah dengan domain Anda. Sebaiknya atur akun ini ke angka yang lebih tinggi.
- Jika Anda mengaktifkan kontrol akses berbutir halus di domain di mana kontrol akses berbutir halus tidak disiapkan, untuk sumber data yang dibuat untuk kueri langsung, Anda perlu mengatur sendiri peran kontrol akses berbutir halus. Untuk informasi selengkapnya tentang cara mengatur peran akses berbutir halus, lihat Membuat integrasi sumber data OpenSearch Layanan Amazon dengan Amazon S3.

Mengubah pengguna utama

Jika Anda lupa rincian pengguna utama, Anda dapat mengonfigurasi ulang menggunakan konsol, AWS CLI, atau API konfigurasi.

Untuk mengubah pengguna utama (konsol)

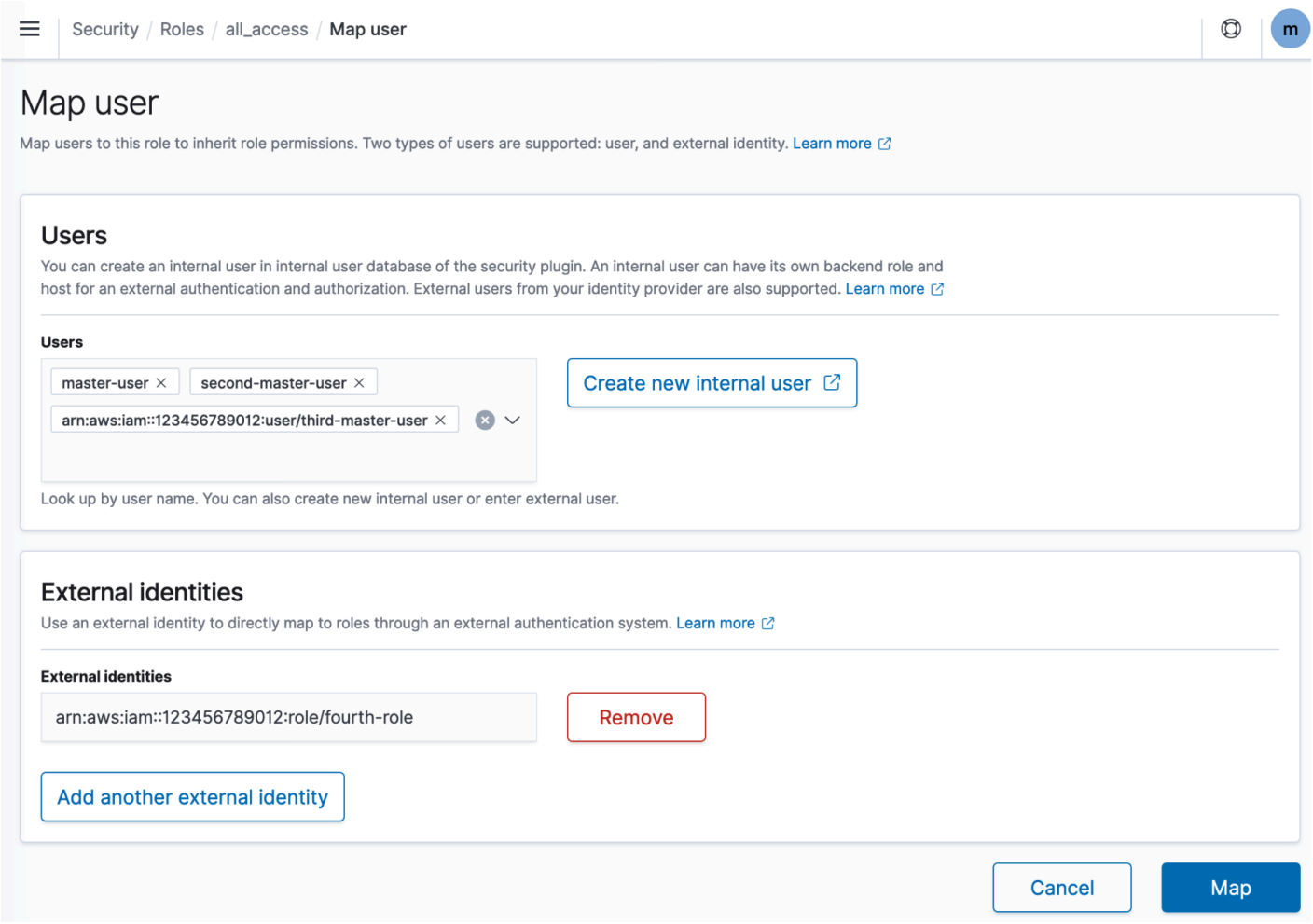
1. Arahkan ke konsol OpenSearch Layanan Amazon di https://console.aws.amazon.com/aos/home/.

2. Pilih domain Anda dan pilih Tindakan, Edit konfigurasi keamanan.
3. Pilih salah satu Set IAM ARN sebagai pengguna master atau Buat pengguna master.
   - Jika sebelumnya Anda menggunakan pengguna utama IAM, kontrol akses detail memetakan ulang peran `all_access` ke ARN IAM baru yang Anda tentukan.
   - Jika Anda sebelumnya menggunakan basis data pengguna internal, kontrol akses detail menciptakan pengguna utama baru. Anda dapat menggunakan pengguna utama baru untuk menghapus yang lama.
   - Beralih dari basis data pengguna internal untuk pengguna utama IAM tidak menghapus pengguna mana pun dari basis data pengguna internal. Sebaliknya, itu hanya menonaktifkan autentikasi basic HTTP. Hapus pengguna secara manual dari basis data pengguna internal, atau menyimpannya jika Anda mungkin perlu mengaktifkan kembali autentikasi basic HTTP.
4. Pilih Simpan perubahan.

Pengguna utama tambahan

Anda menetapkan pengguna utama ketika Anda membuat domain, tetapi jika Anda ingin, Anda dapat menggunakan pengguna utama ini untuk membuat pengguna utama tambahan. Anda memiliki dua opsi: OpenSearch Dasbor atau REST API.

- Di Dasbor, pilih Keamanan, Peran, lalu petakan pengguna master baru ke `security_manager` peran `all_access` dan.



- Untuk menggunakan API REST, kirim permintaan berikut:

```
PUT _plugins/_security/api/rolesmapping/all_access
{
  "backend_roles": [
    "arn:aws:iam::123456789012:role/fourth-master-user"
  ],
  "hosts": [],
  "users": [
    "master-user",
    "second-master-user",
    "arn:aws:iam::123456789012:user/third-master-user"
  ]
}
```

```
PUT _plugins/_security/api/rolesmapping/security_manager
{
```

```
  "backend_roles": [
    "arn:aws:iam::123456789012:role/fourth-master-user"
  ],
  "hosts": [],
  "users": [
    "master-user",
    "second-master-user",
    "arn:aws:iam::123456789012:user/third-master-user"
  ]
}
```

Permintaan ini menggantikan pemetaan peran saat ini, jadi lakukan permintaan GET pertama sehingga Anda dapat menyertakan semua peran saat ini di permintaan PUT. REST API sangat berguna jika Anda tidak dapat mengakses Dasbor dan ingin memetakan peran IAM dari Amazon Cognito ke peran tersebut. `all_access`

Snapshot manual

Kontrol akses detail memperkenalkan beberapa komplikasi tambahan dengan mengambil snapshot manual. Untuk mendaftarkan repositori snapshot — bahkan jika Anda menggunakan autentikasi basic HTTP untuk semua tujuan lain—Anda harus memetakan peran `manage_snapshots` ke IAM role yang memiliki peran izin `iam:PassRole` untuk mengasumsikan `TheSnapshotRole`, seperti yang didefinisikan dalam the section called “Prasyarat”.

Kemudian gunakan IAM role tersebut untuk mengirim permintaan yang ditandatangani ke domain, seperti yang diuraikan dalam the section called “Mendaftarkan repositori snapshot manual”.

Integrasi

Jika Anda menggunakan AWS layanan lain dengan OpenSearch Layanan, Anda harus memberikan peran IAM untuk layanan tersebut dengan izin yang sesuai. Misalnya, aliran pengiriman Firehose sering menggunakan peran IAM yang disebut. `firehose_delivery_role` Di Dasbor, buat peran untuk kontrol akses berbutir halus, dan petakan peran IAM ke sana. Dalam kasus ini, peran baru memerlukan izin berikut:

```
{
  "cluster_permissions": [
    "cluster_composite_ops",
    "cluster_monitor"
```

```
  ],
  "index_permissions": [{
    "index_patterns": [
      "firehose-index*"
    ],
    "allowed_actions": [
      "create_index",
      "manage",
      "crud"
    ]
  }]
}
```

Izin bervariasi berdasarkan tindakan yang dilakukan setiap layanan. AWS IoT Aturan atau AWS Lambda fungsi yang mengindeks data kemungkinan memerlukan izin serupa dengan Firehose, sedangkan fungsi Lambda yang hanya melakukan penelusuran dapat menggunakan set yang lebih terbatas.

Perbedaan API REST

REST API kontrol akses berbutir halus sedikit berbeda tergantung pada versi /Elasticsearch Anda OpenSearch. Sebelum membuat permintaan PUT, buat permintaan GET untuk memverifikasi isi permintaan yang diharapkan. Misalnya, permintaan GET ke `_plugins/_security/api/user` mengembalikan semua pengguna, yang kemudian dapat diubah dan digunakan untuk membuat permintaan PUT yang valid.

Pada Elasticsearch 6.X, permintaan untuk membuat pengguna terlihat seperti ini:

```
PUT _opendistro/_security/api/user/new-user
{
  "password": "some-password",
  "roles": ["new-backend-role"]
}
```

Pada OpenSearch atau Elasticsearch 7.x, permintaan terlihat seperti ini (ubah `_plugins` menjadi `_opendistro` jika menggunakan Elasticsearch):

```
PUT _plugins/_security/api/user/new-user
{
  "password": "some-password",
  "backend_roles": ["new-backend-role"]
```

```
}
```

Selanjutnya, penyewa adalah properti peran dalam Elasticsearch 6.X:

```
GET _opendistro/_security/api/roles/all_access

{
  "all_access": {
    "cluster": ["UNLIMITED"],
    "tenants": {
      "admin_tenant": "RW"
    },
    "indices": {
      "*": {
        "*": ["UNLIMITED"]
      }
    },
    "readonly": "true"
  }
}
```

Di OpenSearch dan Elasticsearch 7.x, mereka adalah objek dengan URI mereka sendiri (ubah `_plugins` ke `_opendistro` if menggunakan Elasticsearch)::

```
GET _plugins/_security/api/tenants

{
  "global_tenant": {
    "reserved": true,
    "hidden": false,
    "description": "Global tenant",
    "static": false
  }
}
```

Untuk dokumentasi tentang OpenSearch REST API, lihat referensi API plugin Keamanan.

> **Tip**
>
> Jika Anda menggunakan basis data pengguna internal, Anda dapat menggunakan curl untuk membuat permintaan dan menguji domain Anda. Coba perintah contoh berikut:

```
curl -XGET -u 'master-user:master-user-password' 'domain-endpoint/_search'
curl -XGET -u 'master-user:master-user-password' 'domain-endpoint/_plugins/
_security/api/user'
```

Tutorial: Konfigurasikan domain dengan pengguna master IAM dan otentikasi Amazon Cognito

Tutorial ini mencakup kasus penggunaan OpenSearch Layanan Amazon yang populer untuk kontrol akses berbutir halus: pengguna master IAM dengan otentikasi Amazon Cognito untuk Dasbor. OpenSearch

Dalam tutorial, kita akan mengkonfigurasi peran master IAM dan peran IAM terbatas, yang kemudian akan kita kaitkan dengan pengguna di Amazon Cognito. Pengguna master kemudian dapat masuk ke OpenSearch Dasbor, memetakan pengguna terbatas ke peran, dan menggunakan kontrol akses berbutir halus untuk membatasi izin pengguna.

Meskipun langkah-langkah ini menggunakan kolam pengguna Amazon Cognito untuk autentikasi, proses dasar yang sama ini bekerja untuk penyedia autentikasi Cognito yang memungkinkan Anda menetapkan peran IAM yang berbeda untuk pengguna yang berbeda.

Anda akan menyelesaikan langkah-langkah berikut dalam tutorial ini:

1. Buat peran master dan IAM terbatas
2. Buat domain dengan otentikasi Cognito
3. Konfigurasikan kumpulan pengguna Cognito dan kumpulan identitas
4. Peran peta di OpenSearch Dasbor

5. Uji izin

Langkah 1: Buat master dan peran IAM terbatas

Arahkan ke konsol AWS Identity and Access Management (IAM) dan buat dua peran terpisah:

- `MasterUserRole`— Pengguna master, yang akan memiliki izin penuh ke cluster dan mengelola peran dan pemetaan peran.
- `LimitedUserRole`— Peran yang lebih terbatas, yang akan Anda berikan akses terbatas sebagai pengguna utama.

Untuk petunjuk membuat peran, lihat Membuat peran menggunakan kebijakan kepercayaan khusus.

Kedua peran harus memiliki kebijakan kepercayaan berikut, yang memungkinkan kumpulan identitas Cognito Anda untuk mengambil peran:

```
{
  "Version": "2012-10-17",
  "Statement": [{
    "Effect": "Allow",
    "Principal": {
      "Federated": "cognito-identity.amazonaws.com"
    },
    "Action": "sts:AssumeRoleWithWebIdentity",
    "Condition": {
      "StringEquals": {
        "cognito-identity.amazonaws.com:aud": "{identity-pool-id}"
      },
      "ForAnyValue:StringLike": {
        "cognito-identity.amazonaws.com:amr": "authenticated"
      }
    }
  }]
}
```

> **Note**
>
> Ganti `identity-pool-id` dengan pengenal unik kumpulan identitas Amazon Cognito Anda. Misalnya, `us-east-1:0c6cdba7-3c3c-443b-a958-fb9feb207aa6`.

Langkah 2: Buat domain dengan otentikasi Cognito

Arahkan ke konsol OpenSearch Layanan Amazon di https://console.aws.amazon.com/aos/home/ dan buat domain dengan pengaturan berikut:

- OpenSearch 1.0 atau yang lebih baru, atau Elasticsearch 7.8 atau yang lebih baru
- Akses publik
- Kontrol akses berbutir halus diaktifkan dengan `MasterUserRole` sebagai pengguna utama (dibuat pada langkah sebelumnya)
- Otentikasi Amazon Cognito diaktifkan untuk Dasbor. OpenSearch Untuk petunjuk untuk mengaktifkan otentikasi Cognito dan memilih kumpulan pengguna dan identitas, lihat. the section called “Mengonfigurasi domain untuk menggunakan otentikasi Amazon Cognito”
- Kebijakan akses domain berikut:

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Effect": "Allow",
      "Principal": {
        "AWS": [
          "arn:aws:iam::{account-id}:role/*"
        ]
      },
      "Action": [
        "es:ESHttp*"
      ],
      "Resource": "arn:aws:es:{region}:{account-id}:domain/{domain-name}/*"
    }
  ]
}
```

- HTTPS diperlukan untuk semua lalu lintas ke domain
- ode-to-node Enkripsi N
- Enkripsi data saat tidak digunakan

Langkah 3: Konfigurasikan pengguna Cognito

Saat domain Anda sedang dibuat, konfigurasikan master dan pengguna terbatas dalam Amazon Cognito dengan mengikuti Buat kumpulan pengguna di Panduan Pengembang Amazon Cognito. Terakhir, konfigurasikan kumpulan identitas Anda dengan mengikuti langkah-langkah di Buat kumpulan identitas di Amazon Cognito. Kumpulan pengguna dan kumpulan identitas harus sama Wilayah AWS.

Langkah 4: Memetakan peran di OpenSearch Dasbor

Setelah pengguna dikonfigurasi, Anda dapat masuk ke OpenSearch Dasbor sebagai pengguna utama dan memetakan pengguna ke peran.

1. Kembali ke konsol OpenSearch Layanan dan arahkan ke URL OpenSearch Dasbor untuk domain yang Anda buat. URL mengikuti format ini:*domain-endpoint*`/_dashboards/`.
2. Masuk dengan `master-user` kredensialnya.
3. Pilih Tambahkan data sampel dan tambahkan contoh data penerbangan.
4. Di panel navigasi kiri, pilih Keamanan, Peran, Buat peran.
5. Beri nama peran `new-role`.
6. Untuk Indeks, tentukan `opensearch_dashboards_sample_data_fli*` (`kibana_sample_data_fli*`pada domain Elasticsearch).
7. Untuk izin Indeks, pilih baca.
8. Untuk keamanan tingkat Dokumen, tentukan kueri berikut:

```
{
  "match": {
    "FlightDelay": true
  }
}
```

9. Untuk keamanan tingkat lapangan, pilih Kecualikan dan tentukan. `FlightNum`
10. Untuk Anonimisasi, tentukan. `Dest`
11. Pilih Buat.
12. Pilih Pengguna yang dipetakan, Kelola pemetaan. Tambahkan Nama Sumber Daya Amazon (ARN) **`LimitedUserRole`** sebagai identitas eksternal dan pilih Peta.

13. Kembali ke daftar peran dan pilih opensearch_dashboards_user. Pilih Pengguna yang Dipetakan, Kelola pemetaan. Tambahkan ARN untuk **`LimitedUserRole`** sebagai peran backend dan pilih Peta.

Langkah 5: Uji izin

Jika peran Anda dipetakan dengan benar, Anda dapat masuk sebagai pengguna terbatas dan menguji izin.

1. Di jendela browser pribadi yang baru, navigasikan ke URL OpenSearch Dasbor untuk domain, masuk menggunakan `limited-user` kredensialnya, dan pilih Jelajahi sendiri.
2. Buka Dev Tools dan jalankan pencarian default:

   ```
   GET _search
   {
     "query": {
       "match_all": {}
     }
   }
   ```

   Perhatikan kesalahan izin. `limited-user` tidak memiliki izin untuk menjalankan pencarian luas klaster.
3. Jalankan pencarian lain:

   ```
   GET opensearch_dashboards_sample_data_flights/_search
   {
     "query": {
       "match_all": {}
     }
   }
   ```

   Perhatikan bahwa semua dokumen yang cocok memiliki `FlightDelay` bidang `true`, bidang `Dest` yang anonim, dan tidak ada bidang `FlightNum`.
4. Di jendela browser asli Anda, masuk sebagai `master-user`, pilih Alat Dev, dan kemudian lakukan pencarian yang sama. Perhatikan perbedaan izin, jumlah klik, dokumen yang cocok, dan bidang yang disertakan.

Tutorial: Konfigurasikan domain dengan database pengguna internal dan otentikasi dasar HTTP

Tutorial ini mencakup kasus penggunaan kontrol akses berbutir halus lainnya yang populer: pengguna utama dalam database pengguna internal dan otentikasi dasar HTTP untuk Dasbor. OpenSearch Pengguna master kemudian dapat masuk ke OpenSearch Dasbor, membuat pengguna internal, memetakan pengguna ke peran, dan menggunakan kontrol akses berbutir halus untuk membatasi izin pengguna.

Anda akan menyelesaikan langkah-langkah berikut dalam tutorial ini:

1. Buat domain dengan pengguna master
2. Konfigurasikan pengguna internal di OpenSearch Dasbor
3. Peran peta di OpenSearch Dasbor
4. Uji izin

Langkah 1: Buat domain

Arahkan ke konsol OpenSearch Layanan Amazon di https://console.aws.amazon.com/aos/home/ dan buat domain dengan pengaturan berikut:

- OpenSearch 1.0 atau yang lebih baru, atau Elasticsearch 7.9 atau yang lebih baru
- Akses publik
- Kontrol akses detail dengan pengguna utama dalam basis data pengguna internal (`TheMasterUser` untuk sisa tutorial ini)
- Otentikasi Amazon Cognito untuk Dasbor dinonaktifkan
- Kebijakan akses berikut:

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Effect": "Allow",
      "Principal": {
        "AWS": [
          "arn:aws:iam::{account-id}:user/*"
        ]
      },
```

```
      "Action": [
        "es:ESHttp*"
      ],
      "Resource": "arn:aws:es:{region}:{account-id}:domain/{domain-name}/*"
    }
  ]
}
```

- HTTPS diperlukan untuk semua lalu lintas ke domain
- ode-to-node Enkripsi N
- Enkripsi data saat tidak digunakan

Langkah 2: Buat pengguna internal di OpenSearch Dasbor

Sekarang setelah Anda memiliki domain, Anda dapat masuk ke OpenSearch Dasbor dan membuat pengguna internal.

1. Kembali ke konsol OpenSearch Layanan dan arahkan ke URL OpenSearch Dasbor untuk domain yang Anda buat. URL mengikuti format ini:*domain-endpoint*/_dashboards/.
2. Masuk dengan`TheMasterUser`.
3. Pilih Tambahkan data sampel dan tambahkan contoh data penerbangan.
4. Di panel navigasi kiri, pilih Keamanan, Pengguna internal, Buat pengguna internal.
5. Beri nama pengguna `new-user` dan tentukan kata sandi. Lalu pilih Buat.

Langkah 3: Memetakan peran di OpenSearch Dasbor

Sekarang setelah pengguna Anda dikonfigurasi, Anda dapat memetakan pengguna Anda ke peran.

1. Tetap di bagian Keamanan OpenSearch Dasbor dan pilih Peran, Buat peran.
2. Beri nama peran `new-role`.
3. Untuk Indeks, tentukan `opensearch_dashboards_sample_data_fli*` (`kibana_sample_data_fli*`pada domain Elasticsearch) untuk pola indeks.
4. Untuk grup tindakan, pilih baca.
5. Untuk keamanan tingkat Dokumen, tentukan kueri berikut:

```
{
  "match": {
```

```
    "FlightDelay": true
  }
}
```

6. Untuk keamanan tingkat lapangan, pilih Kecualikan dan tentukan. `FlightNum`
7. Untuk Anonimisasi, tentukan. `Dest`
8. Pilih Buat.
9. Pilih Pengguna yang dipetakan, Kelola pemetaan. Kemudian tambahkan `new-user` ke Pengguna dan pilihPeta.
10. Kembali ke daftar peran dan pilih opensearch_dashboards_user. Pilih Pengguna yang dipetakan, Kelola pemetaan. Kemudian tambahkan `new-user` ke Pengguna dan pilihPeta.

Langkah 4: Uji izin

Jika peran Anda dipetakan dengan benar, Anda dapat masuk sebagai pengguna terbatas dan menguji izin.

1. Di jendela browser pribadi yang baru, navigasikan ke URL OpenSearch Dasbor untuk domain, masuk menggunakan `new-user` kredensialnya, dan pilih Jelajahi sendiri.
2. Buka Dev Tools dan jalankan pencarian default:

```
GET _search
{
  "query": {
    "match_all": {}
  }
}
```

   Perhatikan kesalahan izin. `new-user` tidak memiliki izin untuk menjalankan pencarian luas klaster.
3. Jalankan pencarian lain:

```
GET dashboards_sample_data_flights/_search
{
  "query": {
    "match_all": {}
  }
}
```

Perhatikan bahwa semua dokumen yang cocok memiliki `FlightDelay` bidang `true`, bidang `Dest` yang anonim, dan tidak ada bidang `FlightNum`.

4. Di jendela browser asli Anda, masuk sebagai `TheMasterUser`, pilih Alat Dev dan lakukan pencarian yang sama. Perhatikan perbedaan izin, jumlah klik, dokumen yang cocok, dan bidang yang disertakan.

Validasi kepatuhan untuk Layanan Amazon OpenSearch

Auditor pihak ketiga menilai keamanan dan kepatuhan OpenSearch Layanan Amazon sebagai bagian dari beberapa program AWS kepatuhan. Hal ini mencakup SOC, PCI, dan HIPAA.

Jika Anda memiliki persyaratan kepatuhan, pertimbangkan untuk menggunakan versi atau Elasticsearch 6.0 OpenSearch atau yang lebih baru. Versi Elasticsearch sebelumnya tidak menawarkan kombinasi enkripsi data saat istirahat dan node-to-node enkripsi dan tidak mungkin memenuhi kebutuhan Anda. Anda juga dapat mempertimbangkan untuk menggunakan versi OpenSearch atau Elasticsearch 6.7 atau yang lebih baru jika kontrol akses halus penting untuk kasus penggunaan Anda. Terlepas dari itu, memilih versi tertentu OpenSearch atau Elasticsearch saat Anda membuat domain tidak menjamin kepatuhan.

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

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

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

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

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

- AWS Sumber Daya AWS — Kumpulan buku kerja dan panduan ini mungkin berlaku untuk industri dan lokasi Anda.
- AWS Panduan Kepatuhan Pelanggan - Memahami model tanggung jawab bersama melalui lensa kepatuhan. Panduan ini merangkum praktik terbaik untuk mengamankan Layanan AWS dan memetakan panduan untuk kontrol keamanan di berbagai kerangka kerja (termasuk Institut Standar dan Teknologi Nasional (NIST), Dewan Standar Keamanan Industri Kartu Pembayaran (PCI), dan Organisasi Internasional untuk Standardisasi (ISO)).
- Mengevaluasi Sumber Daya dengan Aturan dalam Panduan AWS Config Pengembang — AWS Config Layanan menilai seberapa baik konfigurasi sumber daya Anda mematuhi praktik internal, pedoman industri, dan peraturan.
- AWS Security Hub— Ini Layanan AWS memberikan pandangan komprehensif tentang keadaan keamanan Anda di dalamnya AWS. Security Hub menggunakan kontrol keamanan untuk mengevaluasi sumber daya AWS Anda dan memeriksa kepatuhan Anda terhadap standar industri keamanan dan praktik terbaik. Untuk daftar layanan dan kontrol yang didukung, lihat Referensi kontrol Security Hub.
- AWS Audit ManagerIni Layanan AWS membantu Anda terus mengaudit AWS penggunaan Anda untuk menyederhanakan cara Anda mengelola risiko dan kepatuhan terhadap peraturan dan standar industri.

Ketahanan di Amazon OpenSearch Service

Infrastruktur global AWS dibangun di sekitar Wilayah AWS dan Availability Zone. Wilayah AWS menyediakan beberapa Availability Zone yang terpisah secara fisik dan terisolasi yang terhubung dengan jaringan latensi rendah, throughput tinggi, dan jaringan yang sangat berlebihan. Dengan Availability Zone, Anda dapat merancang dan mengoperasikan aplikasi dan basis data yang secara otomatis melakukan failover di antara Availability Zone tanpa gangguan. Availability Zone memiliki ketersediaan yang tinggi, toleran terhadap kesalahan, dan dapat diskalakan jika dibandingkan dengan infrastruktur pusat data tunggal atau ganda tradisional.

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

SelainAWSinfrastruktur global, OpenSearch Service menawarkan beberapa fitur untuk membantu mendukung ketahanan data dan kebutuhan pencadangan Anda:

- Domain Multi-AZ dan serpihan replika
- Snapshot otomatis dan manual

Keamanan infrastruktur di Amazon OpenSearch Service

Sebagai layanan terkelola, Amazon OpenSearch Service 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 OpenSearch Layanan 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 dapat menggunakan AWS Security Token Service (AWS STS) untuk menghasilkan kredensial keamanan sementara untuk menandatangani permintaan.

Anda menggunakan panggilan API yang AWS dipublikasikan untuk mengakses API konfigurasi OpenSearch Layanan melalui jaringan. Untuk mengonfigurasi versi TLS minimum yang diperlukan untuk menerima, tentukan `TLSSecurityPolicy` nilai dalam opsi titik akhir domain:

```
aws opensearch update-domain-config --domain-name my-domain --domain-endpoint-options
 '{"TLSSecurityPolicy": "Policy-Min-TLS-1-2-2019-07"}'
```

Untuk detailnya, lihat referensiAWS CLI perintah.

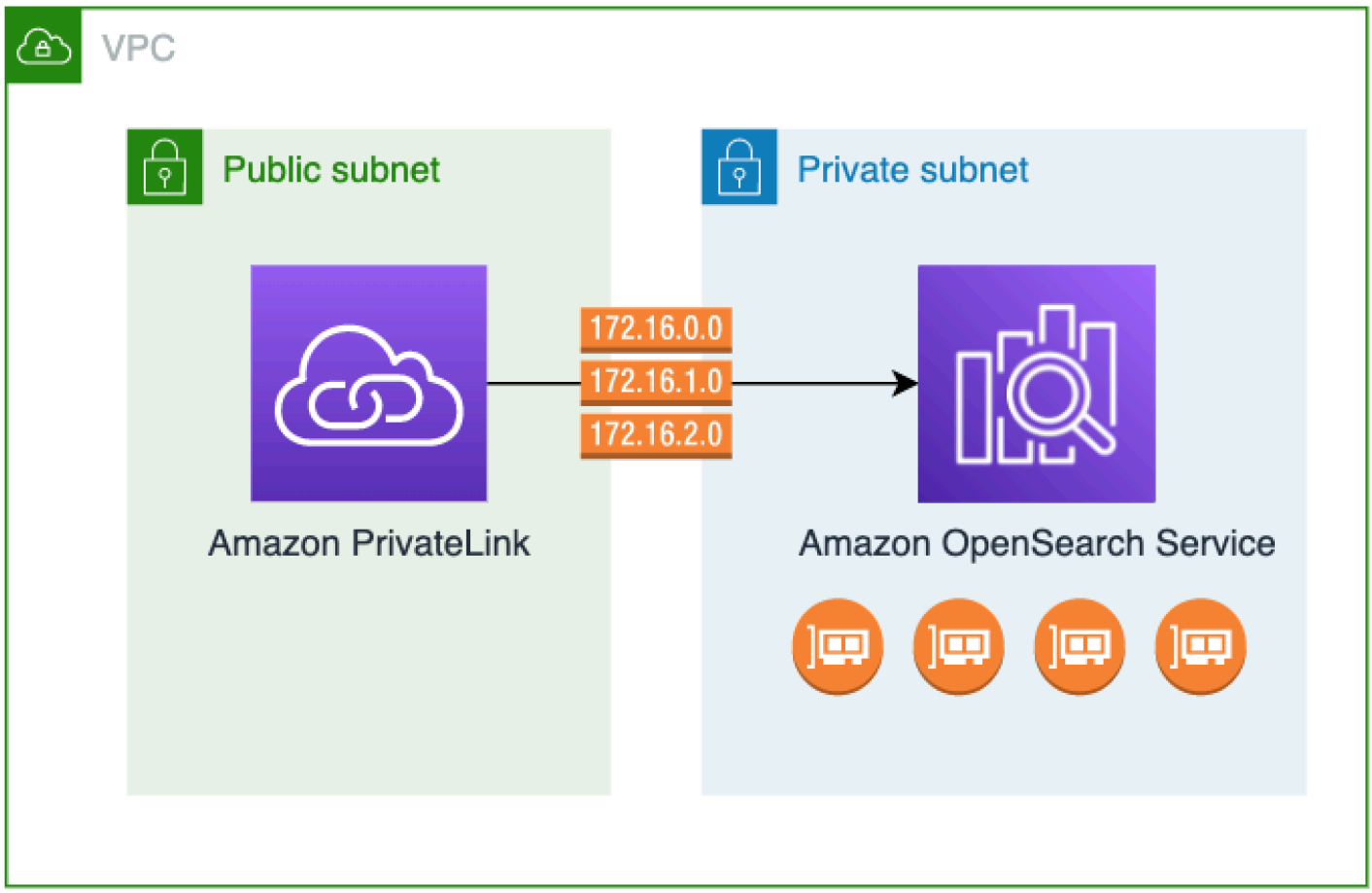
Bergantung pada konfigurasi domain Anda, Anda mungkin juga perlu menandatangani permintaan ke OpenSearch API. Untuk informasi selengkapnya, lihat the section called “Membuat dan menandatangani Permintaan OpenSearch layanan”.

OpenSearch Layanan mendukung domain akses publik, yang dapat menerima permintaan dari perangkat yang terhubung ke internet, dan domain akses VPC, yang terisolasi dari internet publik.

Mengakses OpenSearch Layanan Amazon menggunakan titik akhir OpenSearch VPC yang dikelola Layanan ()AWS PrivateLink

Anda dapat mengakses domain OpenSearch Layanan Amazon dengan menyiapkan titik akhir OpenSearch VPC yang dikelola Layanan (didukung oleh). AWS PrivateLinkTitik akhir ini membuat koneksi pribadi antara VPC Anda dan Layanan Amazon OpenSearch . Anda dapat mengakses domain VPC OpenSearch Layanan seolah-olah berada di VPC Anda, tanpa menggunakan gateway internet, perangkat NAT, koneksi VPN, atau koneksi. AWS Direct Connect Instans di VPC Anda tidak memerlukan alamat IP publik untuk OpenSearch mengakses Layanan.

Anda dapat mengonfigurasi domain OpenSearch Layanan untuk mengekspos titik akhir tambahan yang berjalan pada subnet publik atau pribadi dalam VPC yang sama, VPC yang berbeda, atau yang berbeda. Akun AWSIni memungkinkan Anda menambahkan lapisan keamanan tambahan untuk mengakses domain Anda di mana pun mereka berjalan, tanpa infrastruktur untuk dikelola. Diagram berikut menggambarkan titik akhir VPC yang OpenSearch dikelola Layanan dalam VPC yang sama:



Anda membuat koneksi pribadi ini dengan membuat titik akhir VPC antarmuka yang OpenSearch dikelola Layanan, didukung oleh. AWS PrivateLinkKami membuat antarmuka jaringan endpoint di setiap subnet yang Anda aktifkan untuk titik akhir VPC antarmuka. Ini adalah antarmuka jaringan yang dikelola layanan yang berfungsi sebagai titik masuk untuk lalu lintas yang ditujukan untuk Layanan. OpenSearch Harga endpointAWS PrivateLink antarmuka standar berlaku untuk titik akhir VPC yang OpenSearch dikelola Layanan yang ditagih di bawah. AWS PrivateLink

Anda dapat membuat titik akhir VPC untuk domain yang menjalankan semua versi dan Elasticsearch lama. OpenSearch Untuk informasi selengkapnya, lihat Mengakses Layanan AWS melalui AWS PrivateLink di PanduanAWS PrivateLink .

Pertimbangan dan batasan untuk Layanan OpenSearch

Sebelum Anda menyiapkan titik akhir VPC antarmuka untuk OpenSearch Layanan, tinjau Pertimbangan dalam Panduan.AWS PrivateLink

Saat menggunakan titik akhir OpenSearch VPC yang dikelola Layanan, pertimbangkan hal berikut:

- Anda hanya dapat menggunakan titik akhir VPC antarmuka untuk terhubung ke domain VPC. Domain publik tidak didukung.
- Titik akhir VPC hanya dapat terhubung ke domain dalam hal yang sama. Wilayah AWS
- HTTPS adalah satu-satunya protokol yang didukung untuk titik akhir VPC. HTTP tidak diperbolehkan.
- OpenSearch Layanan mendukung panggilan ke semua operasi OpenSearch API yang didukung melalui titik akhir VPC antarmuka.
- Anda dapat mengonfigurasi maksimum 50 titik akhir per akun, dan maksimum 10 titik akhir per domain. Satu domain dapat memiliki maksimal 10 prinsipal resmi.
- Saat ini Anda tidak dapat menggunakan AWS CloudFormation untuk membuat titik akhir VPC antarmuka.
- Anda hanya dapat membuat titik akhir VPC antarmuka melalui konsol OpenSearch Layanan atau menggunakan API Layanan. OpenSearch Anda tidak dapat membuat titik akhir VPC antarmuka untuk OpenSearch Layanan menggunakan konsol VPC Amazon.
- OpenSearch Titik akhir VPC yang dikelola layanan tidak dapat diakses dari internet. Titik akhir OpenSearch VPC yang dikelola Layanan hanya dapat diakses dalam VPC di mana titik akhir disediakan atau VPC apa pun yang diintip dengan VPC tempat titik akhir disediakan, sebagaimana diizinkan oleh tabel rute dan grup keamanan.
- Kebijakan titik akhir VPC tidak didukung untuk Layanan. OpenSearch Anda dapat mengaitkan grup keamanan dengan antarmuka jaringan titik akhir untuk mengontrol lalu lintas ke OpenSearch Layanan melalui titik akhir VPC antarmuka.
- Peran terkait layanan Anda harus berada di AWS akun yang sama dengan yang Anda gunakan untuk membuat titik akhir VPC.
- Untuk membuat, memperbarui, dan menghapus titik akhir VPC OpenSearch Layanan, Anda harus memiliki izin Amazon EC2 berikut selain izin Layanan Amazon Anda: OpenSearch
  - `ec2:CreateVpcEndpoint`
  - `ec2:DescribeVpcEndpoints`
  - `ec2:ModifyVpcEndpoint`
  - `ec2:DeleteVpcEndpoints`
  - `ec2:CreateTags`
  - `ec2:DescribeTags`
  - `ec2:DescribeSubnets`
  - `ec2:DescribeSecurityGroups`

- `ec2:DescribeVpcs`

> **Note**
>
> Saat ini, Anda tidak dapat membatasi pembuatan titik akhir VPC ke Layanan. OpenSearch Kami sedang berupaya untuk memungkinkan hal ini di pembaruan di masa mendatang.

Memberikan akses ke domain

Jika VPC yang ingin Anda akses domain Anda ada di domain lain Akun AWS, Anda perlu mengotorisasi dari akun pemilik sebelum Anda dapat membuat antarmuka VPC endpoint.

Untuk mengizinkan VPC di tempat lain Akun AWS mengakses domain Anda

1. Buka konsol OpenSearch Layanan Amazon di https://console.aws.amazon.com/aos/home/.
2. Di panel navigasi, pilih Domain dan buka domain yang ingin Anda akses.
3. Buka tab titik akhir VPC, yang menunjukkan akun dan VPC terkait yang memiliki akses ke domain Anda.
4. Pilih Otorisasi Kepala Sekolah.
5. Masukkan Akun AWS ID akun yang akan mengakses domain Anda. Langkah ini mengotorisasi akun yang ditentukan untuk membuat titik akhir VPC terhadap domain.
6. Pilih Izinkan.

Buat antarmuka VPC endpoint untuk domain VPC

Anda dapat membuat titik akhir VPC antarmuka untuk OpenSearch Layanan menggunakan konsol OpenSearch Layanan atau (). AWS Command Line Interface AWS CLI

Untuk membuat titik akhir VPC antarmuka untuk domain Layanan OpenSearch

1. Buka konsol OpenSearch Layanan Amazon di https://console.aws.amazon.com/aos/home/.
2. Di panel navigasi kiri, pilih titik akhir VPC.
3. Pilih Buat titik akhir.
4. Pilih apakah akan menghubungkan domain di saat ini Akun AWS atau yang lain Akun AWS.

5. Pilih domain yang Anda sambungkan dengan titik akhir ini. Jika domain saat ini Akun AWS, gunakan dropdown untuk memilih domain. Jika domain berada di akun yang berbeda, masukkan Nama Sumber Daya Amazon (ARN) dari domain yang akan disambungkan. Untuk memilih domain di akun yang berbeda, pemilik perlu memberi Anda akses ke domain tersebut.
6. Untuk VPC, pilih VPC dari mana Anda akan mengakses Layanan. OpenSearch
7. Untuk Subnet, pilih satu atau beberapa subnet dari mana Anda akan mengakses OpenSearch Layanan.
8. Untuk grup Keamanan, pilih grup keamanan untuk diasosiasikan dengan antarmuka jaringan titik akhir. Ini adalah langkah penting di mana Anda membatasi port, protokol, dan sumber apa untuk lalu lintas masuk yang Anda otorisasi ke titik akhir Anda. Aturan grup keamanan harus mengizinkan sumber daya yang akan menggunakan titik akhir VPC untuk berkomunikasi dengan OpenSearch Layanan untuk berkomunikasi dengan antarmuka jaringan titik akhir.
9. Pilih Buat titik akhir. Titik akhir harus aktif dalam 2-5 menit.

Bekerja dengan titik akhir OpenSearch VPC yang dikelola Layanan menggunakan API konfigurasi

Gunakan operasi API berikut untuk membuat dan mengelola titik akhir OpenSearch VPC yang dikelola Layanan.

- CreateVpcEndpoint
- ListVpcEndpoints
- UpdateVpcEndpoint
- DeleteVpcEndpoint

Gunakan operasi API berikut untuk mengelola akses titik akhir ke domain VPC:

- AuthorizeVpcEndpointAccess
- ListVpcEndpointAccess
- ListVpcEndpointsForDomain
- RevokeVpcEndpointAccess

Otentikasi SAMP untuk Dasbor OpenSearch

Autentikasi SAMP untuk OpenSearch Dasbor memungkinkan Anda menggunakan penyedia identitas yang ada untuk menawarkan sistem masuk tunggal (SSO) untuk Dasbor di domain OpenSearch Layanan Amazon yang berjalan atau Elasticsearch 6.7 atau yang lebih baru. OpenSearch Untuk menggunakan autentikasi SAML, Anda harus mengaktifkan Kontrol akses detail.

Daripada mengautentikasi melalui Amazon Cognito atau database pengguna internal, autentikasi SAMP OpenSearch untuk Dasbor memungkinkan Anda menggunakan penyedia identitas pihak ketiga untuk masuk ke Dasbor, mengelola kontrol akses berbutir halus, mencari data, dan membangun visualisasi. OpenSearch Layanan mendukung penyedia yang menggunakan standar SAMP 2.0, seperti Okta, Keycloak, Active Directory Federation Services (ADFS), Auth0, dan. AWS IAM Identity Center

Otentikasi SAMP untuk Dasbor hanya untuk mengakses OpenSearch Dasbor melalui browser web. Kredensyal SAMP Anda tidak memungkinkan Anda membuat permintaan HTTP langsung ke API OpenSearch atau Dasbor.

Gambaran umum konfigurasi SAML

Dokumentasi ini mengasumsikan bahwa Anda memiliki penyedia identitas yang ada dan beberapa keakraban dengannya. Kami tidak dapat memberikan langkah-langkah konfigurasi terperinci untuk penyedia Anda yang tepat, hanya untuk domain OpenSearch Layanan Anda.

Alur login OpenSearch Dasbor dapat mengambil salah satu dari dua bentuk:

- Penyedia layanan (SP) dimulai: Anda menavigasi ke Dasbor (misalnya,`https://my-domain.us-east-1.es.amazonaws.com/_dashboards`), yang mengarahkan Anda ke layar login. Setelah Anda masuk, penyedia identitas mengarahkan Anda ke Dasbor.
- Penyedia identitas (iDP) dimulai: Anda menavigasi ke penyedia identitas, masuk, dan memilih OpenSearch Dasbor dari direktori aplikasi.

OpenSearch Layanan menyediakan dua URL masuk tunggal, SP-initiated dan IDP-initiated, tetapi Anda hanya memerlukan satu yang sesuai dengan alur login Dashboard yang Anda inginkan. OpenSearch

Apa pun jenis autentikasi yang Anda gunakan, tujuannya adalah untuk login melalui penyedia identitas dan menerima pernyataan SAML yang berisi nama pengguna Anda (wajib) dan peran

backend (opsional, tetapi direkomendasikan). Informasi ini memungkinkan kontrol akses detail untuk menetapkan izin bagi pengguna SAML. Dalam penyedia identitas eksternal, peran backend biasanya disebut “peran” atau “grup”.

Pertimbangan

Pertimbangkan hal berikut ketika Anda mengkonfigurasi otentikasi SAMP:

- Karena ukuran file metadata iDP, kami sangat menyarankan menggunakan AWS konsol untuk mengonfigurasi otentikasi SAMP.
- Domain hanya mendukung satu metode otentikasi Dasbor pada satu waktu. Jika Anda mengaktifkan otentikasi Amazon Cognito untuk OpenSearch Dasbor, Anda harus menonaktifkannya sebelum dapat mengaktifkan otentikasi SAMP.
- Jika Anda menggunakan penyeimbang beban jaringan dengan SAMP, Anda harus terlebih dahulu membuat endpoint kustom. Untuk informasi selengkapnya, lihat ???.

Otentikasi SAMP untuk domain VPC

SAMP tidak memerlukan komunikasi langsung antara penyedia identitas Anda dan penyedia layanan Anda. Oleh karena itu, bahkan jika OpenSearch domain Anda di-host dalam VPC pribadi, Anda masih dapat menggunakan SAMP selama browser Anda dapat berkomunikasi dengan OpenSearch cluster dan penyedia identitas Anda. Browser Anda pada dasarnya bertindak sebagai perantara antara penyedia identitas Anda dan penyedia layanan Anda. Untuk diagram berguna yang menjelaskan alur otentikasi SAMP, lihat dokumentasi Okta.

Memodifikasi kebijakan akses domain

Sebelum Anda mengkonfigurasi otentikasi SAMP, Anda harus memperbarui kebijakan akses domain untuk memungkinkan pengguna SAMP mengakses domain. Jika tidak, Anda akan melihat kesalahan akses ditolak.

Kami merekomendasikan kebijakan akses domain berikut, yang menyediakan akses penuh ke subresource (`/*`) pada domain:

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

```
    {
      "Effect": "Allow",
      "Principal": {
        "AWS": "*"
      },
      "Action": "es:ESHttp*",
      "Resource": "domain-arn/*"
    }
  ]
}
```

Untuk membuat kebijakan lebih ketat, Anda dapat menambahkan kondisi alamat IP ke kebijakan. Kondisi ini membatasi akses hanya ke rentang alamat IP atau subnet yang ditentukan. Misalnya, kebijakan berikut hanya mengizinkan akses dari subnet 192.0.2.0/24:

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Effect": "Allow",
      "Principal": {
        "AWS": "*"
      },
      "Action": [
        "es:ESHttp*"
      ],
      "Condition": {
        "IpAddress": {
          "aws:SourceIp": [
            "192.0.2.0/24"
          ]
        }
      },
      "Resource": "domain-arn/*"
    }
  ]
}
```

> **Note**
>
> Kebijakan akses domain terbuka memerlukan kontrol akses berbutir halus untuk diaktifkan di domain Anda, jika tidak, Anda akan melihat kesalahan berikut:

`To protect domains with public access, a restrictive policy or fine-grained access control is required.`
Jika Anda memiliki pengguna utama atau pengguna internal yang dikonfigurasi dengan kata sandi yang kuat, menjaga kebijakan tetap terbuka saat menggunakan kontrol akses berbutir halus mungkin dapat diterima dari sudut pandang keamanan. Untuk informasi selengkapnya, lihat ???.

Mengkonfigurasi otentikasi yang diprakarsai SP- atau IDP

Langkah-langkah ini menjelaskan cara mengaktifkan otentikasi SAMP dengan otentikasi yang diprakarsai SP atau IDP untuk Dasbor. OpenSearch Untuk langkah tambahan yang diperlukan untuk mengaktifkan keduanya, lihat Mengonfigurasi otentikasi yang diprakarsai SP dan IDP.

Langkah 1: Aktifkan otentikasi SAMP

Anda dapat mengaktifkan otentikasi SAMP baik selama pembuatan domain, atau dengan memilih Tindakan, Edit konfigurasi keamanan pada domain yang ada. Langkah-langkah berikut sedikit berbeda tergantung pada mana yang Anda pilih.

Dalam konfigurasi domain, di bawah otentikasi SAMP untuk OpenSearch Dasbor/Kibana, pilih Aktifkan otentikasi SAMP.

Langkah 2: Konfigurasikan penyedia identitas Anda

Lakukan langkah-langkah berikut tergantung pada saat Anda mengonfigurasi otentikasi SAMP.

Jika Anda membuat domain baru

Jika Anda sedang dalam proses membuat domain baru, OpenSearch Layanan belum dapat menghasilkan ID entitas penyedia layanan atau URL SSO. Penyedia identitas Anda memerlukan nilai-nilai ini untuk mengaktifkan otentikasi SAMP dengan benar, tetapi mereka hanya dapat dihasilkan setelah domain dibuat. Untuk mengatasi saling ketergantungan ini selama pembuatan domain, Anda dapat memberikan nilai sementara ke dalam konfigurasi iDP Anda untuk menghasilkan metadata yang diperlukan dan kemudian memperbaruinya setelah domain Anda aktif.

Jika Anda menggunakan endpoint kustom, Anda dapat menyimpulkan seperti apa URL-nya. Misalnya, jika titik akhir kustom Anda, ID entitas penyedia layanan akan menjadi`www.custom-endpoint.com`, URL SSO yang diprakarsai IDP akan menjadi `www.custom-`

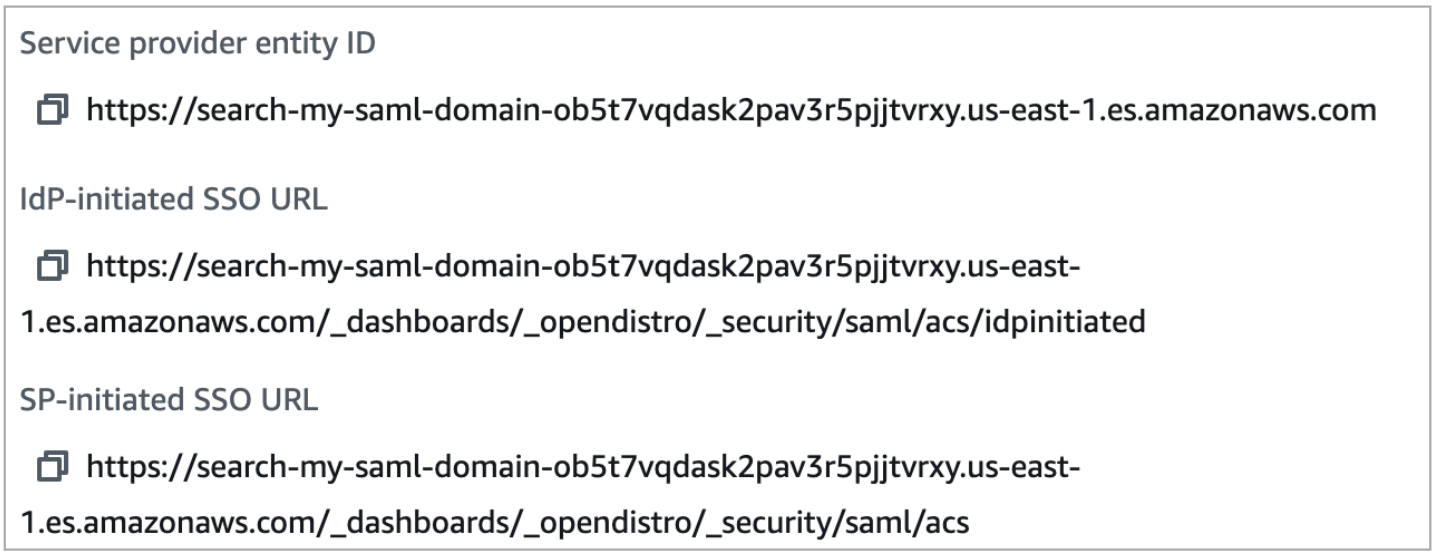
endpoint.comwww.*custom-endpoint*.com/_dashboards/_opendistro/_security/saml/acs/idpinitiated, dan URL SSO yang diprakarsai SP akan menjadi. www.*custom-endpoint*.com/_dashboards/_opendistro/_security/saml/acs Anda dapat menggunakan nilai untuk mengonfigurasi penyedia identitas Anda sebelum domain dibuat. Lihat bagian selanjutnya untuk contoh.

Jika Anda tidak menggunakan titik akhir kustom, Anda dapat memasukkan nilai sementara ke dalam iDP untuk menghasilkan metadata yang diperlukan, lalu memperbaruinya nanti setelah domain aktif.

Misalnya, dalam Okta, Anda dapat masuk https://*temp-endpoint*.amazonaws.com ke kolom Single sign on URL dan Audience URI (SP Entity ID), yang memungkinkan Anda menghasilkan metadata. Kemudian, setelah domain aktif, Anda dapat mengambil nilai yang benar dari OpenSearch Layanan dan memperbaruinya di Okta. Untuk petunjuk, lihat the section called “Langkah 6: Perbarui URL IDP Anda”.

Jika Anda mengedit domain yang ada

Jika Anda mengaktifkan otentikasi SAMP pada domain yang ada, salin ID entitas penyedia layanan dan salah satu URL SSO. Untuk panduan tentang URL mana yang akan digunakan, lihatthe section called “Gambaran umum konfigurasi SAML”.



Gunakan nilai untuk mengonfigurasi penyedia identitas Anda. Bagian ini merupakan proses yang paling rumit, dan sayangnya, terminologi dan langkah-langkah sangat bervariasi berdasarkan penyedia. Baca dokumentasi dari penyedia Anda.

Di Okta, misalnya, Anda membuat aplikasi web SAMP 2.0. Untuk URL tanda tunggal pada URL, tentukan URL SSO. Untuk URI audiens (ID Entitas SP, tentukan ID entitas SP.

Okta memiliki pengguna dan grup, bukan pengguna dan peran backend. Untuk Pernyataan Atribut Grup, kami sarankan Anda menambahkan `role` ke bidang Nama dan ekspresi reguler `.+` ke bidang Filter. Pernyataan ini memberi tahu penyedia identitas Okta untuk memasukkan semua grup pengguna pada bidang `role` dari penegasan SAML setelah pengguna mengautentikasi.

Di Pusat Identitas IAM, Anda menentukan ID entitas SP sebagai audiens SAMP Aplikasi. Anda juga perlu menentukan pemetaan atribut berikut: `Subject=${user:name}` dan. `Role=${user:groups}`

Di Auth0, Anda membuat aplikasi web biasa dan mengaktifkan add-on SAMP 2.0. Di Keycloak, Anda membuat klien.

Langkah 3: Impor metadata IDP

Setelah Anda konfigurasi, penyedia identitas akan menghasilkan file metadata IdP. File XHTML ini berisi informasi tentang penyedia, seperti sertifikat TLS, titik akhir masuk tunggal, dan ID entitas penyedia identitas.

Salin konten file metadata iDP dan tempelkan ke bidang Metadata dari iDP di konsol Layanan. OpenSearch Sebagai alternatif, pilih Impor dari file XHTML dan unggah file. File metadata harus terlihat seperti ini:

```
<?xml version="1.0" encoding="UTF-8"?>
<md:EntityDescriptor entityID="entity-id"
 xmlns:md="urn:oasis:names:tc:SAML:2.0:metadata">
  <md:IDPSSODescriptor WantAuthnRequestsSigned="false"
 protocolSupportEnumeration="urn:oasis:names:tc:SAML:2.0:protocol">
    <md:KeyDescriptor use="signing">
      <ds:KeyInfo xmlns:ds="http://www.w3.org/2000/09/xmldsig#">
        <ds:X509Data>
          <ds:X509Certificate>tls-certificate</ds:X509Certificate>
        </ds:X509Data>
      </ds:KeyInfo>
    </md:KeyDescriptor>
    <md:NameIDFormat>urn:oasis:names:tc:SAML:1.1:nameid-format:unspecified</
md:NameIDFormat>
    <md:NameIDFormat>urn:oasis:names:tc:SAML:1.1:nameid-format:emailAddress</
md:NameIDFormat>
    <md:SingleSignOnService Binding="urn:oasis:names:tc:SAML:2.0:bindings:HTTP-POST"
 Location="idp-sso-url"/>
    <md:SingleSignOnService Binding="urn:oasis:names:tc:SAML:2.0:bindings:HTTP-
Redirect" Location="idp-sso-url"/>
```

```
  </md:IDPSSODescriptor>
</md:EntityDescriptor>
```

Langkah 4: Konfigurasikan bidang SAMP

Setelah memasukkan metadata IDP, konfigurasikan bidang tambahan berikut dalam konsol Layanan: OpenSearch

- ID entitas IDP — Salin nilai `entityID` properti dari file metadata Anda dan tempelkan ke bidang ini. Banyak penyedia identitas juga menampilkan nilai ini sebagai bagian dari ringkasan pasca-konfigurasi. Beberapa penyedia menyebutnya “penerbit”.
- Nama pengguna master SAMP dan peran backend master SAMP — Peran pengguna dan/atau backend yang Anda tentukan menerima izin penuh ke cluster, setara dengan pengguna master baru, tetapi hanya dapat menggunakan izin tersebut di dalam Dasbor. OpenSearch

  Di Okta, misalnya, Anda mungkin memiliki pengguna `jdoe` yang termasuk dalam grup `admins`. Jika Anda menambahkan `jdoe` ke bidang Nama pengguna utama SAML, hanya pengguna yang akan menerima izin penuh. Jika Anda menambahkan `admins` ke bidang peran backend master SAMP, setiap pengguna yang termasuk dalam `admins` grup akan menerima izin penuh.

  > **Note**
  >
  > Isi pernyataan SAMP harus sama persis dengan string yang Anda gunakan untuk nama pengguna master SAMP dan peran master SAMP. Beberapa penyedia identitas menambahkan awalan sebelum nama pengguna mereka, yang dapat menyebabkan ketidakcocokan hard-to-diagnose . Di antarmuka pengguna penyedia identitas, Anda mungkin melihat `jdoe`, tetapi penegasan SAML mungkin berisi `auth0|jdoe`. Selalu gunakan string dari penegasan SAML.

Banyak penyedia identitas mengizinkan Anda melihat contoh penegasan selama proses konfigurasi, dan alat-alat seperti SAML-tracer dapat membantu Anda memeriksa dan memecahkan masalah konten penegasan yang sesungguhnya. Penegasan terlihat seperti ini:

```
<?xml version="1.0" encoding="UTF-8"?>
<saml2:Assertion ID="id67229299299259351343340162"
 IssueInstant="2020-09-22T22:03:08.633Z" Version="2.0"
  xmlns:saml2="urn:oasis:names:tc:SAML:2.0:assertion">
```

```
  <saml2:Issuer Format="urn:oasis:names:tc:SAML:2.0:nameid-format:entity">idp-issuer</
saml2:Issuer>
  <saml2:Subject>
    <saml2:NameID Format="urn:oasis:names:tc:SAML:1.1:nameid-
format:unspecified">username</saml2:NameID>
    <saml2:SubjectConfirmation Method="urn:oasis:names:tc:SAML:2.0:cm:bearer">
      <saml2:SubjectConfirmationData NotOnOrAfter="2020-09-22T22:08:08.816Z"
 Recipient="domain-endpoint/_dashboards/_opendistro/_security/saml/acs"/>
    </saml2:SubjectConfirmation>
  </saml2:Subject>
  <saml2:Conditions NotBefore="2020-09-22T21:58:08.816Z"
 NotOnOrAfter="2020-09-22T22:08:08.816Z">
    <saml2:AudienceRestriction>
      <saml2:Audience>domain-endpoint</saml2:Audience>
    </saml2:AudienceRestriction>
  </saml2:Conditions>
  <saml2:AuthnStatement AuthnInstant="2020-09-22T19:54:37.274Z">
    <saml2:AuthnContext>

 <saml2:AuthnContextClassRef>urn:oasis:names:tc:SAML:2.0:ac:classes:PasswordProtectedTransport<
saml2:AuthnContextClassRef>
    </saml2:AuthnContext>
  </saml2:AuthnStatement>
  <saml2:AttributeStatement>
    <saml2:Attribute Name="role" NameFormat="urn:oasis:names:tc:SAML:2.0:attrname-
format:unspecified">
      <saml2:AttributeValue
        xmlns:xs="http://www.w3.org/2001/XMLSchema"
        xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
 xsi:type="xs:string">GroupName Match Matches regex ".+" (case-sensitive)
      </saml2:AttributeValue>
    </saml2:Attribute>
  </saml2:AttributeStatement>
</saml2:Assertion>
```

Langkah 5: (Opsional) Konfigurasikan pengaturan tambahan

Di bawah Pengaturan tambahan, konfigurasikan bidang opsional berikut:

- Kunci subjek - Anda dapat membiarkan bidang ini kosong untuk menggunakan `NameID` elemen pernyataan SAMP untuk nama pengguna. Jika penegasan Anda tidak menggunakan elemen standar ini dan sebagai gantinya menyertakan nama pengguna sebagai atribut kustom, tentukan atribut di sini.

- Kunci peran - Jika Anda ingin menggunakan peran backend (disarankan), tentukan atribut dari pernyataan di bidang ini, seperti atau. `role group` Ini adalah situasi berbeda yang memungkinkan alat seperti SAML-tracer untuk membantu Anda
- Waktu sesi untuk hidup - Secara default, OpenSearch Dasbor mengeluarkan pengguna setelah 24 jam. Anda dapat mengonfigurasi nilai ini ke angka apa pun antara 60 dan 1.440 (24 jam) dengan menentukan nilai baru.

Setelah Anda puas dengan konfigurasi Anda, simpan domain.

Langkah 6: Perbarui URL IDP Anda

Jika Anda mengaktifkan otentikasi SAMP saat membuat domain, Anda harus menentukan URL sementara dalam IDP Anda untuk menghasilkan file metadata XHTML. Setelah status domain berubah`Active`, Anda bisa mendapatkan URL yang benar dan memodifikasi IDP Anda.

Untuk mengambil URL, pilih domain dan pilih Tindakan, Edit konfigurasi keamanan. Di bawah otentikasi SAMP untuk OpenSearch Dasbor/Kibana, Anda dapat menemukan ID entitas penyedia layanan dan URL SSO yang benar. Salin nilai dan gunakan untuk mengonfigurasi penyedia identitas Anda, menggantikan URL sementara yang Anda berikan di langkah 2.

Langkah 7: Memetakan pengguna SAMP ke peran

Setelah status domain Anda Aktif dan idP Anda dikonfigurasi dengan benar, navigasikan ke OpenSearch Dasbor.

- Jika Anda memilih URL yang diinisiasi SP, buka *`domain-endpoint`*`/_dashboards`. Untuk masuk ke penyewa tertentu secara langsung, Anda dapat menambahkan `?security_tenant=`*`tenant-name`* ke URL.
- Jika Anda memilih URL yang diinisiasi IDP, buka direktori aplikasi penyedia identitas Anda.

Dalam kedua kasus, login sebagai pengguna utama SAML atau pengguna yang termasuk dalam peran backend utama SAML. Untuk melanjutkan contoh dari langkah 7, login sebagai `jdoe` atau anggota grup `admins`.

Setelah OpenSearch Dasbor dimuat, pilih Keamanan, Peran. Kemudian, petakan peran untuk memungkinkan pengguna lain mengakses OpenSearch Dasbor.

Misalnya, Anda dapat memetakan rekan Anda yang tepercaya `jroe` ke peran `all_access` dan `security_manager`. Anda juga dapat memetakan peran backend `analysts` ke peran `readall` dan `opensearch_dashboards_user`.

Jika Anda lebih suka menggunakan API daripada OpenSearch Dasbor, lihat contoh permintaan berikut:

```
PATCH _plugins/_security/api/rolesmapping
[
  {
    "op": "add", "path": "/security_manager", "value": { "users": ["master-user",
 "jdoe", "jroe"], "backend_roles": ["admins"] }
  },
  {
    "op": "add", "path": "/all_access", "value": { "users": ["master-user", "jdoe",
 "jroe"], "backend_roles": ["admins"] }
  },
  {
    "op": "add", "path": "/readall", "value": { "backend_roles": ["analysts"] }
  },
  {
    "op": "add", "path": "/opensearch_dashboards_user", "value": { "backend_roles":
 ["analysts"] }
  }
]
```

Mengkonfigurasi otentikasi yang diprakarsai SP dan IDP

Jika ingin mengonfigurasi autentikasi SP yang diinisiasi dan IDP yang diinisiasi Anda harus melakukannya melalui penyedia identitas. Misalnya, di Okta, Anda dapat melakukan langkah-langkah berikut:

1. Dalam aplikasi SAMP Anda, pergi ke General, pengaturan SAMP.
2. Untuk URL tanda Tunggal di URL, berikan URL SSO yang diprakarsai IDP Anda. Misalnya, `https://search-`*`domain-hash`*`/_dashboards/_opendistro/_security/saml/acs/`*`idpinitiated`*.
3. Aktifkan Izinkan aplikasi ini untuk meminta URL SSO lainnya.
4. Di bawah URL SSO yang Dapat Diminta, tambahkan satu atau beberapa URL SSO yang diprakarsai SP. Misalnya, `https://search-`*`domain-hash`*`/_dashboards/_opendistro/_security/saml/`*`acs`*.

Mengkonfigurasi otentikasi SAMP ()AWS CLI

AWS CLI Perintah berikut memungkinkan otentikasi SAMP untuk OpenSearch Dasbor pada domain yang ada:

```
aws opensearch update-domain-config \
  --domain-name my-domain \
  --advanced-security-options '{"SAMLOptions":{"Enabled":true,"MasterUserName":"my-idp-user","MasterBackendRole":"my-idp-group-or-role","Idp":{"EntityId":"entity-id","MetadataContent":"metadata-content-with-quotes-escaped"},"RolesKey":"optional-roles-key","SessionTimeoutMinutes":180,"SubjectKey":"optional-subject-key"}}'
```

Anda harus mengeluarkan semua tanda kutip dan karakter baris baru dalam XML metadata. Misalnya, gunakan `<KeyDescriptor use=\"signing\">\n` sebagai ganti `<KeyDescriptor use="signing">` dan pemisah baris. Untuk informasi rinci tentang penggunaan AWS CLI, lihat Referensi AWS CLI Perintah.

Mengkonfigurasi otentikasi SAMP (API konfigurasi)

Permintaan berikut ke API konfigurasi memungkinkan otentikasi SAMP untuk OpenSearch Dasbor pada domain yang ada:

```
POST https://es.us-east-1.amazonaws.com/2021-01-01/opensearch/domain/my-domain/config
{
  "AdvancedSecurityOptions": {
    "SAMLOptions": {
      "Enabled": true,
      "MasterUserName": "my-idp-user",
      "MasterBackendRole": "my-idp-group-or-role",
      "Idp": {
        "EntityId": "entity-id",
        "MetadataContent": "metadata-content-with-quotes-escaped"
      },
      "RolesKey": "optional-roles-key",
      "SessionTimeoutMinutes": 180,
      "SubjectKey": "optional-subject-key"
    }
  }
}
```

Anda harus mengeluarkan semua tanda kutip dan karakter baris baru dalam XML metadata. Misalnya, gunakan `<KeyDescriptor use=\"signing\">\n` sebagai ganti `<KeyDescriptor use="signing">` dan pemisah baris. Untuk informasi mendetail tentang penggunaan API konfigurasi, lihat referensi API OpenSearch Layanan.

Memecahkan masalah SAML

Kesalahan	Detail
Permintaan Anda: '*/some/path* ' tidak diizinkan.	Verifikasi bahwa Anda telah memberikan URL SSO yang benar (langkah 3) ke penyedia identitas.
Harap berikan dokumen metadata penyedia identitas yang valid untuk mengaktifkan SAMP.	File metadata IdP Anda tidak sesuai dengan standar SAML 2.0. Periksa kesalahan menggunakan alat validasi.
Opsi konfigurasi SAML tidak terlihat di konsol.	Perbarui ke perangkat lunak layanan terbaru.
Kesalahan konfigurasi SAMP: Ada yang tidak beres saat mengambil konfigurasi SAMP, silakan periksa pengaturan Anda.	Kesalahan umum ini dapat terjadi karena berbagai alasan. • Pastikan Anda telah memberikan ID entitas SP dan URL SSO yang benar kepada penyedia identitas Anda. • Buat ulang file metadata IdP, dan verifikasi ID entitas IdP tersebut. Tambahkan metadata yang telah diperbarui ke dalam konsol AWS . • Verifikasi bahwa kebijakan akses domain Anda memungkinkan akses ke OpenSearch Dasbor dan`_plugins/_security/*` . Secara umum, kami merekomendasikan kebijakan akses terbuka untuk domain yang menggunakan kontrol akses detail. • Baca dokumentasi penyedia identitas Anda untuk mengetahui langkah-langkah dalam mengonfigurasi SAML.

Kesalahan	Detail
Peran tidak ada: Tidak ada peran yang tersedia untuk pengguna ini, silakan hubungi administrator sistem Anda.	Anda berhasil diautentikasi, tetapi nama pengguna dan setiap peran backend dari penegasan SAML tidak dipetakan ke peran apa pun sehingga tidak memiliki izin. Pemetaan ini peka huruf besar dan kecil. Administrator sistem Anda dapat memverifikasi konten pernyataan SAMP Anda menggunakan alat seperti SAML-tracer, dan kemudian memeriksa pemetaan peran Anda menggunakan permintaan berikut: `GET _plugins/_security/api/rolesmapping`
Browser Anda terus mengalihkan atau menerima kesalahan HTTP 500 saat mencoba mengakses OpenSearch Dasbor.	Kesalahan ini dapat terjadi jika penegasan SAML berisi peran yang berjumlah sekitar 1.500 karakter. Misalnya, jika Anda meneruskan 80 peran dengan panjang rata-rata adalah 20 karakter, Anda mungkin melebihi batas ukuran cookie di peramban web Anda. Dimulai dengan OpenSearch versi 2.7, pernyataan SAMP mendukung peran hingga 5000 karakter.
Anda tidak dapat keluar dari ADFS.	ADFS mengharuskan semua permintaan logout ditandatangani, yang tidak didukung oleh OpenSearch Layanan. Hapus `<SingleLogoutService />` dari file metadata IDP untuk memaksa OpenSearch Service menggunakan mekanisme logout internalnya sendiri.
`Could not find entity descriptor for __PATH__.`	ID entitas dari IDP yang disediakan dalam metadata OpenSearch XML-Service berbeda dari yang ada di respon SAMP. Untuk memperbaikinya, pastikan mereka cocok. Aktifkan log Kesalahan Aplikasi CW di domain Anda untuk menemukan pesan kesalahan untuk men-debug masalah integrasi SAMP.

Kesalahan	Detail
`Signature validation failed. SAML response rejected.`	OpenSearch Layanan tidak dapat memverifikasi tanda tangan dalam respons SAMP menggunak an sertifikat IDP yang disediakan dalam metadata XHTML. Ini bisa berupa kesalahan manual, atau IDP Anda telah memutar sertifikatnya. Perbarui sertifikat terbaru dari IDP Anda dalam metadata XHTML yang disediakan untuk Layanan melalui. OpenSearch AWS Management Console
`__PATH__ is not a valid audience for this response.`	Bidang audiens dalam respons SAMP tidak cocok dengan titik akhir domain. Untuk memperbaiki kesalahan ini, perbarui bidang audiens SP agar sesuai dengan titik akhir domain Anda. Jika Anda telah mengaktifkan titik akhir kustom, bidang audiens harus sesuai dengan titik akhir kustom Anda. Aktifkan log Kesalahan Aplikasi CW di domain Anda untuk menemukan pesan kesalahan untuk men-debug masalah integrasi SAMP.
Browser Anda menerima kesalahan HTTP 400 dengan `Invalid Request Id` dalam respons.	Kesalahan ini umumnya terjadi jika Anda telah mengonfigurasi URL yang diprakarsai IDP dengan format. *`<DashboardsURL>`* `/_opendistro/_security/saml/acs` Sebagai gantinya, konfigurasikan URL dengan format*`<DashboardsURL>`* `/_opendistro/_security/saml/acs/idpinitiated` .

Kesalahan	Detail
Tanggapan diterima di __PATH__ bukannya__PATH__.	Bidang tujuan dalam respons SAMP tidak cocok dengan salah satu format URL berikut: • *<DashboardsURL>* /_opendistro/_security/saml/acs • *<DashboardsURL>* /_opendistro/_security/saml/acs/idpinitiated . Bergantung pada alur masuk yang Anda gunakan (dimulai SP atau dimulai IDP), masukkan bidang tujuan yang cocok dengan salah satu URL. OpenSearch
Respons memiliki InResponseTo atribut, sementara tidak InResponseTo diharapkan.	Anda menggunakan URL yang diprakarsai IDP untuk alur login yang dimulai SP. Gunakan URL yang diprakarsai SP sebagai gantinya.

Mengaktifkan autentikasi SAML

Untuk menonaktifkan otentikasi SAMP untuk OpenSearch Dasbor (konsol)

1. Pilih domain, Tindakan, dan Edit konfigurasi keamanan.
2. Hapus tanda centang pada opsi Aktifkan autentikasi SAML.
3. Pilih Simpan perubahan.
4. Setelah domain selesai diproses, verifikasi pemetaan peran kontrol akses berbutir halus dengan permintaan berikut:

   ```
   GET _plugins/_security/api/rolesmapping
   ```

   Menonaktifkan otentikasi SAMP untuk Dasbor tidak menghapus pemetaan untuk nama pengguna master SAMP dan/atau peran backend master SAMP. Jika Anda ingin menghapus pemetaan ini, masuk ke Dasbor menggunakan database pengguna internal (jika diaktifkan), atau gunakan API untuk menghapusnya:

```
PUT _plugins/_security/api/rolesmapping/all_access
{
  "users": [
    "master-user"
  ]
}
```

Mengonfigurasi otentikasi Amazon Cognito untuk Dasbor OpenSearch

Anda dapat mengautentikasi dan melindungi instalasi OpenSearch Dasbor default OpenSearch Layanan Amazon Anda menggunakan Amazon Cognito. Otentikasi Amazon Cognito bersifat opsional dan hanya tersedia untuk domain yang menggunakan OpenSearch atau Elasticsearch 5.1 atau yang lebih baru. Jika Anda tidak mengonfigurasi autentikasi Amazon Cognito, Anda masih dapat melindungi Dasbor menggunakan kebijakan akses berbasis IP dan server proxy, otentikasi dasar HTTP, atau SALL.

Sebagian besar proses otentikasi terjadi di Amazon Cognito, tetapi bagian ini menawarkan pedoman dan persyaratan untuk mengonfigurasi sumber daya Amazon Cognito agar berfungsi dengan domain Layanan. OpenSearch Harga standar berlaku untuk semua sumber daya Amazon Cognito.

> **Tip**
>
> Pertama kali Anda mengonfigurasi domain untuk menggunakan otentikasi Amazon Cognito untuk OpenSearch Dasbor, sebaiknya gunakan konsol. Sumber daya Amazon Cognito sangat disesuaikan, dan konsol dapat membantu Anda mengidentifikasi dan memahami fitur yang penting bagi Anda.

Topik

Prasyarat

Sebelum Anda dapat mengonfigurasi otentikasi Amazon Cognito untuk OpenSearch Dasbor, Anda harus memenuhi beberapa prasyarat. Konsol OpenSearch Layanan membantu merampingkan pembuatan sumber daya ini, tetapi memahami tujuan setiap sumber daya membantu konfigurasi dan pemecahan masalah. Autentikasi Amazon Cognito untuk Dasbor memerlukan sumber daya berikut:

- Kolam pengguna Amazon Cognito
- Kolam identitas Amazon Cognito
- IAM role yang memiliki kebijakan `AmazonOpenSearchServiceCognitoAccess` dilampirkan (`CognitoAccessForAmazonOpenSearch`)

> **Note**
>
> Kumpulan pengguna dan kumpulan identitas harus samaWilayah AWS. Anda dapat menggunakan kumpulan pengguna, kumpulan identitas, dan peran IAM yang sama untuk menambahkan autentikasi Amazon Cognito untuk Dasbor ke beberapa domain Layanan. OpenSearch Untuk mempelajari selengkapnya, lihat the section called “Quotas”.

Tentang kolam pengguna

Kolam pengguna memiliki dua fitur utama: membuat dan mengelola direktori pengguna, dan memungkinkan pengguna untuk mendaftar dan masuk. Untuk petunjuk cara membuat kumpulan pengguna, lihat Membuat Kumpulan Pengguna di Panduan Pengembang Amazon Cognito.

Saat Anda membuat kumpulan pengguna untuk digunakan dengan OpenSearch Layanan, pertimbangkan hal berikut:

- Kolam pengguna Amazon Cognito Anda harus memiliki nama domain. OpenSearch Layanan menggunakan nama domain ini untuk mengarahkan pengguna ke halaman login untuk mengakses Dasbor. Selain nama domain, kolam pengguna tidak memerlukan konfigurasi non-default.
- Anda harus menentukan atribut standar kolam yang diperlukan—atribut seperti nama, tanggal lahir, alamat email, dan nomor telepon. Anda tidak dapat mengubah atribut ini setelah membuat kolam pengguna, jadi pilih atribut yang penting bagi Anda saat ini.
- Saat membuat kolam pengguna Anda, pilih apakah pengguna dapat membuat akun mereka sendiri, kekuatan sandi minimum untuk akun, dan apakah akan mengaktifkan autentikasi multi-faktor. Jika Anda berencana untuk menggunakan penyedia identitas eksternal, pengaturan ini tidak penting. Secara teknis, Anda dapat mengaktifkan kolam pengguna sebagai penyedia identitas dan mengaktifkan penyedia identitas eksternal, tetapi kebanyakan orang lebih memilih satu atau yang lain.

ID kolam pengguna mengambil bentuk *region*_*ID*. Jika Anda berencana untuk menggunakan AWS CLI atau AWS SDK untuk mengkonfigurasi OpenSearch Layanan, catat ID.

Tentang kolam identitas

Kolam identitas memungkinkan Anda menetapkan peran hak istimewa sementara dan terbatas kepada pengguna setelah mereka masuk. Untuk petunjuk tentang cara membuat kolam identitas, lihat Kumpulan Pengguna di Panduan Developer Amazon Cognito. Saat Anda membuat kumpulan identitas untuk digunakan dengan OpenSearch Layanan, pertimbangkan hal berikut:

- Jika Anda menggunakan konsol Amazon Cognito, Anda harus memilih Aktifkan akses ke identitas yang tidak terautentikasi kotak centang untuk membuat kolam identitas. Setelah Anda membuat kumpulan identitas dan mengonfigurasi domain OpenSearch Layanan, Amazon Cognito menonaktifkan setelan ini.
- Anda tidak perlu menambahkan Penyedia identitas eksternal ke kolam identitas. Saat Anda mengonfigurasi OpenSearch Layanan untuk menggunakan autentikasi Amazon Cognito, layanan akan mengonfigurasi kumpulan identitas untuk menggunakan kumpulan pengguna yang baru saja Anda buat.
- Setelah Anda membuat kolam identitas, Anda harus memilih IAM role yang tidak terautentikasi dan dikonfirmasi. Peran ini menentukan kebijakan akses yang dimiliki pengguna sebelum dan sesudah mereka masuk. Jika Anda menggunakan konsol Amazon Cognito, maka dapat membuat peran ini untuk Anda. Setelah Anda membuat peran yang diautentikasi, membuat

catatan dari ARN, yang mengambil bentuk `arn:aws:iam::123456789012:role/Cognito_identitypoolnameAuth_Role`.

ID kolam identitas mengambil bentuk *region*`:`*ID*`-`*ID*`-`*ID*`-`*ID*`-`*ID*. Jika Anda berencana untuk menggunakan AWS CLI atau AWS SDK untuk mengkonfigurasi OpenSearch Layanan, catat ID.

Tentang peran CognitoAccessForAmazonOpenSearch

OpenSearch Layanan memerlukan izin untuk mengonfigurasi pengguna Amazon Cognito dan kumpulan identitas dan menggunakannya untuk otentikasi. Anda dapat menggunakan`AmazonOpenSearchServiceCognitoAccess`, yang merupakan kebijakan AWS - managed, untuk tujuan ini. `AmazonESCognitoAccess`adalah kebijakan warisan yang digantikan oleh `AmazonOpenSearchServiceCognitoAccess` ketika layanan diubah namanya menjadi Amazon OpenSearch Service. Kedua kebijakan memberikan izin Amazon Cognito minimum yang diperlukan untuk mengaktifkan otentikasi Cognito. Untuk kebijakan JSON, lihat konsol IAM.

Jika Anda menggunakan konsol untuk membuat atau mengonfigurasi domain OpenSearch Layanan, itu akan membuat peran IAM untuk Anda dan melampirkan `AmazonOpenSearchServiceCognitoAccess` kebijakan (atau `AmazonESCognitoAccess` kebijakan jika itu adalah domain Elasticsearch) ke peran tersebut. Nama default untuk peran ini adalah `CognitoAccessForAmazonOpenSearch`.

Kebijakan izin peran `AmazonOpenSearchServiceCognitoAccess` dan `AmazonESCognitoAccess` keduanya memungkinkan OpenSearch Layanan untuk menyelesaikan tindakan berikut pada semua identitas dan kumpulan pengguna:

- Tindakan: `cognito-idp:DescribeUserPool`
- Tindakan: `cognito-idp:CreateUserPoolClient`
- Tindakan: `cognito-idp:DeleteUserPoolClient`
- Tindakan: `cognito-idp:UpdateUserPoolClient`
- Tindakan: `cognito-idp:DescribeUserPoolClient`
- Tindakan: `cognito-idp:AdminInitiateAuth`
- Tindakan: `cognito-idp:AdminUserGlobalSignOut`
- Tindakan: `cognito-idp:ListUserPoolClients`
- Tindakan: `cognito-identity:DescribeIdentityPool`
- Tindakan: `cognito-identity:SetIdentityPoolRoles`

- Tindakan: `cognito-identity:GetIdentityPoolRoles`

Jika Anda menggunakan AWS CLI atau salah satu AWS SDK, Anda harus membuat peran Anda sendiri, melampirkan kebijakan, dan menentukan ARN untuk peran ini ketika OpenSearch Anda mengonfigurasi domain Layanan. Peran harus memiliki hubungan kepercayaan berikut:

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

Untuk instruksi, lihat Membuat Peran untuk Mendelegasikan izin ke sebuah Layanan AWS dan Melampirkan dan Melepaskan Kebijakan IAM di Panduan Pengguna IAM.

Mengonfigurasi domain untuk menggunakan otentikasi Amazon Cognito

Setelah menyelesaikan prasyarat, Anda dapat mengonfigurasi domain OpenSearch Layanan untuk menggunakan Amazon Cognito untuk Dasbor.

> **Note**
>
> Amazon Cognito tidak tersedia di semua. Wilayah AWS Untuk daftar Wilayah yang didukung, lihat Wilayah AWSdan Titik Akhir. Anda tidak perlu menggunakan Wilayah yang sama untuk Amazon Cognito yang Anda gunakan untuk OpenSearch Layanan.

Mengonfigurasi autentikasi Amazon Cognito (konsol)

Karena menciptakan CognitoAccessForAmazonOpenSearchperan untuk Anda, konsol menawarkan pengalaman konfigurasi paling sederhana. Selain izin OpenSearch Layanan standar, Anda memerlukan kumpulan izin berikut untuk menggunakan konsol guna membuat domain yang menggunakan autentikasi Amazon Cognito untuk Dasbor. OpenSearch

```
{
  "Version": "2012-10-17",
  "Statement": [{
      "Effect": "Allow",
      "Action": [
        "ec2:DescribeVpcs",
        "cognito-identity:ListIdentityPools",
        "cognito-idp:ListUserPools",
        "iam:CreateRole",
        "iam:AttachRolePolicy"
      ],
      "Resource": "*"
    },
    {
      "Effect": "Allow",
      "Action": [
        "iam:GetRole",
        "iam:PassRole"
      ],
      "Resource": "arn:aws:iam::123456789012:role/service-
role/CognitoAccessForAmazonOpenSearch"
    }
  ]
}
```

Untuk petunjuk menambahkan izin ke identitas (pengguna, grup pengguna, atau peran), lihat Menambahkan izin identitas IAM (konsol).

Jika `CognitoAccessForAmazonOpenSearch` sudah ada, Anda memerlukan lebih sedikit izin:

```
{
  "Version": "2012-10-17",
  "Statement": [{
      "Effect": "Allow",
      "Action": [
        "ec2:DescribeVpcs",
        "cognito-identity:ListIdentityPools",
        "cognito-idp:ListUserPools"
      ],
      "Resource": "*"
    },
    {
      "Effect": "Allow",
```

```
      "Action": [
        "iam:GetRole",
        "iam:PassRole"
      ],
      "Resource": "arn:aws:iam::123456789012:role/service-
role/CognitoAccessForAmazonOpenSearch"
    }
  ]
}
```

Untuk mengonfigurasi otentikasi Amazon Cognito untuk Dasbor (konsol)

1. Buka konsol OpenSearch Layanan Amazon di https://console.aws.amazon.com/aos/home/.
2. Di bawah Domain, pilih domain yang ingin Anda konfigurasikan.
3. Pilih Tindakan, Edit konfigurasi keamanan.
4. Pilih Aktifkan otentikasi Amazon Cognito.
5. Untuk Wilayah, pilih Wilayah AWS yang berisi kumpulan pengguna Amazon Cognito dan kumpulan identitas.
6. Untuk kumpulan pengguna Cognito, pilih kumpulan pengguna atau buat satu. Untuk panduan, lihat the section called “Tentang kolam pengguna”.
7. Untuk kumpulan identitas Cognito, pilih kumpulan identitas atau buat. Untuk panduan, lihat the section called “Tentang kolam identitas”.

   > **Note**
   >
   > Tautan Buat kumpulan pengguna dan Buat kumpulan identitas mengarahkan Anda ke konsol Amazon Cognito dan mengharuskan Anda membuat sumber daya ini secara manual. Prosesnya tidak otomatis. Untuk mempelajari selengkapnya, lihat the section called “Prasyarat”.

8. Untuk nama peran IAM, gunakan nilai default `CognitoAccessForAmazonOpenSearch` (disarankan) atau masukkan nama baru. Untuk mempelajari selengkapnya tentang tujuan peran ini, lihat the section called “Tentang peran CognitoAccessForAmazonOpenSearch”.
9. Pilih Save changes (Simpan perubahan).

Setelah domain selesai diproses, lihat the section called “Mengizinkan peran terautentikasi” dan the section called “Mengonfigurasi penyedia identitas” untuk langkah-langkah konfigurasi tambahan.

Mengonfigurasi autentikasi Amazon Cognito (AWS CLI)

Gunakan `--cognito-options` parameter untuk mengonfigurasi domain OpenSearch Layanan Anda. Sintaks berikut digunakan oleh perintah `create-domain` dan `update-domain-config`:

```
--cognito-options Enabled=true,UserPoolId="user-pool-id",IdentityPoolId="identity-pool-id",RoleArn="arn:aws:iam::123456789012:role/CognitoAccessForAmazonOpenSearch"
```

Contoh

Contoh berikut membuat domain di `us-east-1` Wilayah yang memungkinkan otentikasi Amazon Cognito untuk Dasbor menggunakan `CognitoAccessForAmazonOpenSearch` peran dan menyediakan akses domain ke: `Cognito_Auth_Role`

```
aws opensearch create-domain --domain-name my-domain --region us-east-1 --access-policies '{ "Version":"2012-10-17", "Statement":[{"Effect":"Allow","Principal":{"AWS":
 ["arn:aws:iam::123456789012:role/
Cognito_Auth_Role"]},"Action":"es:ESHttp*","Resource":"arn:aws:es:us-east-1:123456789012:domain/*" }]}' --engine-version "OpenSearch_1.0"
 --cluster-config InstanceType=m4.xlarge.search,InstanceCount=1
 --ebs-options EBSEnabled=true,VolumeSize=10 --cognito-options
 Enabled=true,UserPoolId="us-east-1_123456789",IdentityPoolId="us-east-1:12345678-1234-1234-1234-123456789012",RoleArn="arn:aws:iam::123456789012:role/
CognitoAccessForAmazonOpenSearch"
```

Setelah domain selesai diproses, lihat the section called “Mengizinkan peran terautentikasi” dan the section called “Mengonfigurasi penyedia identitas” untuk langkah-langkah konfigurasi tambahan.

Mengonfigurasi Autentikasi Amazon Cognito (SDK AWS)

AWSSDK (kecuali SDK Android dan iOS) mendukung semua operasi yang ditentukan dalam Referensi API OpenSearch Layanan Amazon, termasuk `CognitoOptions` parameter untuk `CreateDomain` dan `UpdateDomainConfig` operasi. Untuk informasi selengkapnya tentang menginstal dan menggunakan SDK AWS, lihat AWSKit Pengembangan Perangkat Lunak.

Setelah domain selesai diproses, lihat the section called “Mengizinkan peran terautentikasi” dan the section called “Mengonfigurasi penyedia identitas” untuk langkah-langkah konfigurasi tambahan.

Mengizinkan peran terautentikasi

Secara default, peran IAM terautentikasi yang Anda konfigurasikan dengan mengikuti pedoman di the section called “Tentang kolam identitas” tidak memiliki hak istimewa yang diperlukan untuk mengakses Dasbor. OpenSearch Anda harus memberikan peran dengan izin tambahan.

> **Note**
>
> Jika Anda mengonfigurasi kontrol akses berbutir halus dan menggunakan kebijakan akses terbuka atau berbasis IP, Anda dapat melewati langkah ini.

Anda dapat menyertakan izin ini dalam kebijakan berbasis identitas, tetapi kecuali Anda ingin pengguna yang diautentikasi memiliki akses ke semua domain OpenSearch Layanan, kebijakan berbasis sumber daya yang dilampirkan ke satu domain adalah pendekatan yang lebih baik.

Untuk`Principal`, tentukan ARN dari peran terautentikasi Cognito yang Anda konfigurasikan dengan pedoman. the section called “Tentang kolam identitas”

```
{
   "Version":"2012-10-17",
   "Statement":[
      {
         "Effect":"Allow",
         "Principal":{
            "AWS":[
               "arn:aws:iam::123456789012:role/Cognito_identitypoolnameAuth_Role"
            ]
         },
         "Action":[
            "es:ESHttp*"
         ],
         "Resource":"arn:aws:es:region:123456789012:domain/domain-name/*"
      }
   ]
}
```

Untuk petunjuk tentang menambahkan kebijakan berbasis sumber daya ke domain OpenSearch Layanan, lihat. the section called “Mengonfigurasi kebijakan akses”

Mengonfigurasi penyedia identitas

Saat Anda mengonfigurasi domain untuk menggunakan autentikasi Amazon Cognito untuk Dasbor, OpenSearch Layanan menambahkan klien aplikasi ke kumpulan pengguna dan menambahkan kumpulan pengguna ke kumpulan identitas sebagai penyedia autentikasi.

> **Warning**
>
> Jangan mengganti nama atau menghapus klien aplikasi.

Tergantung pada bagaimana Anda mengonfigurasi kolam pengguna, Anda mungkin perlu membuat akun pengguna secara manual, atau pengguna mungkin dapat membuatnya sendiri. Jika pengaturan ini dapat diterima, Anda tidak perlu melakukan tindakan lebih lanjut. Banyak orang, bagaimanapun, lebih memilih untuk menggunakan penyedia identitas eksternal.

Untuk mengaktifkan penyedia identitas SAML 2.0, Anda harus menyediakan dokumen metadata SAML. Untuk mengaktifkan penyedia identitas sosial seperti Login with Amazon, Facebook, dan Google, Anda harus memiliki ID aplikasi dan rahasia aplikasi dari penyedia layanan tersebut. Anda dapat mengaktifkan kombinasi penyedia identitas.

Cara termudah untuk mengonfigurasi kolam pengguna Anda adalah dengan menggunakan konsol Amazon Cognito. Untuk instruksi, lihat Menggunakan Federasi dari Kolam Pengguna dan Menentukan Pengaturan Penyedia Identitas untuk Aplikasi Kolam Pengguna Anda di Panduan Developer Amazon Cognito.

(Opsional) Mengonfigurasi akses terperinci

Anda mungkin telah memperhatikan bahwa pengaturan kolam identitas default menetapkan setiap pengguna yang mencatat dalam IAM role yang sama (`Cognito_`*`identitypool`*`Auth_Role`), yang berarti bahwa setiap pengguna dapat mengakses sumber daya AWS yang sama. Jika Anda ingin menggunakan kontrol akses detail dengan Amazon Cognito—misalnya, jika Anda ingin analis organisasi Anda memiliki akses hanya-baca ke beberapa indeks, tetapi developer memiliki akses tulis ke semua indeks—Anda memiliki dua opsi:

- Buat grup pengguna dan konfigurasikan penyedia identitas Anda untuk memilih IAM role berdasarkan token autentikasi pengguna (disarankan).
- Konfigurasikan penyedia identitas Anda untuk memilih IAM role berdasarkan satu atau beberapa aturan.

Untuk panduan yang mencakup kontrol akses detail, lihat the section called “Tutorial: Kontrol akses berbutir halus dengan otentikasi Cognito”.

> **Important**
>
> Sama seperti peran default, Amazon Cognito harus menjadi bagian dari hubungan kepercayaan setiap peran tambahan ini. Untuk detailnya, lihat Membuat Peran untuk Pemetaan Peran di Panduan Developer Amazon Cognito.

Kelompok pengguna dan token

Bila Anda membuat grup pengguna, Anda memilih IAM role untuk anggota grup. Untuk informasi tentang membuat grup, lihat Grup Pengguna di Panduan Developer Amazon Cognito.

Setelah Anda membuat satu grup pengguna atau lebih, Anda dapat mengonfigurasi penyedia autentikasi Anda untuk menetapkan peran pengguna grup mereka daripada peran default kolam identitas. Pilih Pilih peran dari token, lalu pilih Gunakan peran default yang Diautentikasi atau DENY untuk menentukan cara kumpulan identitas menangani pengguna yang bukan bagian dari grup.

Aturan

Aturan pada dasarnya adalah serangkaian pernyataan `if` bahwa Amazon Cognito mengevaluasi secara berurutan. Misalnya, jika alamat email pengguna berisi `@corporate`, Amazon Cognito menetapkan pengguna tersebut `Role_A`. Jika alamat email pengguna berisi `@subsidiary`, maka menetapkan pengguna tersebut `Role_B`. Jika tidak, maka menetapkan pengguna menjadi peran terautentikasi default.

Untuk mempelajari selengkapnya, lihat Menggunakan Pemetaan Berbasis Aturan untuk Menetapkan Peran untuk Pengguna di Panduan Developer Amazon Cognito.

(Opsional) Menyesuaikan halaman masuk

Anda dapat menggunakan konsol Amazon Cognito untuk mengunggah logo khusus dan membuat perubahan CSS ke halaman masuk. Untuk petunjuk dan daftar lengkap properti CSS, lihat Menentukan Pengaturan Kustomisasi UI Aplikasi untuk Kolam Pengguna Anda di Panduan Developer Amazon Cognito.

(Opsional) Mengonfigurasi keamanan tingkat lanjut

Kolam pengguna Amazon Cognito mendukung fitur keamanan canggih seperti autentikasi multi-faktor, pemeriksaan kredensial yang dikompromikan, dan autentikasi adaptif. Untuk mempelajari selengkapnya, lihat Mengelola Keamanan di Panduan Developer Amazon Cognito.

Pengujian

Setelah Anda puas dengan konfigurasi Anda, verifikasi bahwa pengalaman pengguna memenuhi harapan Anda.

Untuk mengakses OpenSearch Dasbor

1. Arahkan ke `https://`*`opensearch-domain`*`/_dashboards` di browser web. Untuk masuk ke penyewa tertentu secara langsung, tambahkan `?security_tenant=`*`tenant-name`* ke URL.
2. Masuk menggunakan kredensial pilihan Anda.
3. Setelah OpenSearch Dasbor dimuat, konfigurasikan setidaknya satu pola indeks. Dasbor menggunakan pola-pola ini untuk mengidentifikasi indeks mana yang ingin Anda analisis. Masukkan `*`, pilih Langkah selanjutnya, lalu pilih Buat pola indeks.
4. Untuk mencari atau menjelajahi data Anda, pilih Temukan.

Jika ada langkah dari proses ini yang gagal, lihat the section called “Masalah konfigurasi umum” untuk informasi pemecahan masalah.

Quotas

Amazon Cognito memiliki batas lunak pada banyak sumber dayanya. Jika Anda ingin mengaktifkan otentikasi Dasbor untuk sejumlah besar domain OpenSearch Layanan, tinjau Kuota di Amazon Cognito dan minta kenaikan batas seperlunya.

Setiap domain OpenSearch Layanan menambahkan klien aplikasi ke kumpulan pengguna, yang menambahkan penyedia autentikasi ke kumpulan identitas. Jika Anda mengaktifkan autentikasi OpenSearch Dasbor untuk lebih dari 10 domain, Anda mungkin menemukan batas “penyedia kumpulan pengguna Amazon Cognito maksimum per kumpulan identitas”. Jika Anda melebihi batas, domain OpenSearch Layanan apa pun yang Anda coba konfigurasikan untuk menggunakan autentikasi Amazon Cognito untuk Dasbor dapat macet dalam status konfigurasi Pemrosesan.

Masalah konfigurasi umum

Tabel berikut mencantumkan masalah konfigurasi umum dan solusinya.

Konfigurasi Layanan OpenSearch

Isu	Solusi
`OpenSearch Service can't create the role` (konsol)	Anda tidak memiliki izin IAM yang benar. Tambahkan izin yang ditentukan di the section called “Mengonfigurasi autentikasi Amazon Cognito (konsol)”.
`User is not authorized to perform: iam:PassRole on resource CognitoAccessForAmazonOpenSearch` (konsol)	Anda tidak memiliki `iam:PassRole` izin untuk CognitoAccessForAmazonOpenSearchperan tersebut. Lampirkan kebijakan berikut ke akun Anda: `{ "Version": "2012-10-17", "Statement": [ { "Effect": "Allow", "Action": [ "iam:PassRole" ], "Resource": "arn:aws:iam:: 123456789012:role/service-role/CognitoAccessForAmazonOpenSearch " } ] }` Cara lainnya, Anda dapat melampirkan kebijakan `IAMFullAccess` .
`User is not authorized to perform: cognito-identity:ListIdentityPools on resource`	Anda tidak memiliki izin baca untuk Amazon Cognito. Lampirkan kebijakan `AmazonCognitoReadOnly` untuk akun Anda.
`An error occurred (ValidationException) when calling`	OpenSearch Layanan tidak ditentukan dalam hubungan kepercayaan `CognitoAccessForAmazonOpenS`

Isu	Solusi
`the CreateDomain operation : OpenSearch Service must be allowed to use the passed role`	`earch` peran. Periksa apakah peran Anda menggunak an hubungan kepercayaan yang ditentukan dalam the section called “Tentang peran CognitoAccessForAm azonOpenSearch”. Cara lainnya, gunakan konsol untuk mengonfigurasi autentikasi Amazon Cognito. Konsol membuat peran untuk Anda.
`An error occurred (Validati onException) when calling the CreateDomain operation : User is not authorize d to perform: cognito-i dp: action on resource: user pool`	Peran yang ditentukan di `--cognito-options` tidak memiliki izin untuk mengakses Amazon Cognito. Periksa bahwa peran memiliki kebijakan AWS yang dikelola `AmazonOpenSearchServiceCognitoAccess` yang dilampirkan. Cara lainnya, gunakan konsol untuk mengonfigurasi autentikasi Amazon Cognito. Konsol membuat peran untuk Anda.
`An error occurred (Validati onException) when calling the CreateDomain operation : User pool does not exist`	OpenSearch Layanan tidak dapat menemukan kumpulan pengguna. Konfirmasi bahwa Anda membuat satu dan memiliki ID yang benar. Untuk menemukan ID, Anda dapat menggunakan konsol Amazon Cognito atau perintah AWS CLI berikut: `aws cognito-idp list-user-pools --max-results 60 --region region`
`An error occurred (Validati onException) when calling the CreateDomain operation : IdentityPool not found`	OpenSearch Layanan tidak dapat menemukan kumpulan identitas. Konfirmasi bahwa Anda membuat satu dan memiliki ID yang benar. Untuk menemukan ID, Anda dapat menggunakan konsol Amazon Cognito atau perintah AWS CLI berikut: `aws cognito-identity list-identity-pools --max-results 60 --region region`

Isu	Solusi
`An error occurred (ValidationException) when calling the CreateDomain operation: Domain needs to be specified for user pool`	Kolam pengguna tidak memiliki nama domain. Anda dapat mengonfigurasi satu menggunakan konsol Amazon Cognito atau perintah AWS CLI berikut: `aws cognito-idp create-user-pool-domain --domain name --user-pool-id id`

Mengakses Dasbor OpenSearch

Isu	Solusi
Halaman masuk tidak menampilkan penyedia identitas pilihan saya.	Periksa apakah Anda mengaktifkan penyedia identitas untuk klien aplikasi OpenSearch Layanan seperti yang ditentukan dalamthe section called “Mengonfigurasi penyedia identitas”.
Halaman masuk tidak terlihat seolah-olah terkait dengan organisasi saya.	Lihat the section called “(Opsional) Menyesuaikan halaman masuk”.
Kredensial masuk saya tidak bekerja.	Periksa bahwa Anda telah mengonfigurasi penyedia identitas sebagaimana ditentukan dalam the section called “Mengonfigurasi penyedia identitas”. Jika Anda menggunakan kumpulan pengguna sebagai penyedia identitas, periksa apakah akun tersebut ada di konsol Amazon Cognito.
OpenSearch Dasbor tidak memuat sama sekali atau tidak berfungsi dengan baik.	Peran yang diautentikasi Amazon Cognito memerlukan `es:ESHttp*` izin untuk domain (`/*`) untuk mengakses dan menggunakan Dasbor. Periksa bahwa Anda menambahkan kebijakan akses sebagaimana ditentukan dalam the section called “Mengizinkan peran terautentikasi”.
Ketika saya keluar dari OpenSearch Dasbor dari satu tab, tab yang tersisa	Saat Anda keluar dari sesi OpenSearch Dasbor saat menggunakan autentikasi Amazon Cognito OpenSearc

Isu	Solusi
menampilkan pesan yang menyataka n bahwa token penyegaran telah dicabut.	h , Layanan menjalankan operasi, AdminUserGlobalSig nOutyang membuat Anda keluar dari semua OpenSearch sesi Dasbor aktif.
`Invalid identity pool configuration. Check assigned IAM roles for this pool.`	Amazon Cognito tidak memiliki izin untuk menganggap IAM role atas nama pengguna terautentikasi. Memodifik asi hubungan kepercayaan untuk peran guna menyertak an: `{ "Version": "2012-10-17", "Statement": [{ "Effect": "Allow", "Principal": { "Federated": "cognito-identity. amazonaws.com" }, "Action": "sts:AssumeRoleWithWebIdent ity", "Condition": { "StringEquals": { "cognito-identity.amazonaws.com:aud" : " identity-pool-id " }, "ForAnyValue:StringLike": { "cognito-identity.amazonaws.com:amr" : "authenticated" } } }] }`
`Token is not from a supported provider of this identity pool.`	Kesalahan ini jarang terjadi saat Anda menghapus klien aplikasi dari kolam pengguna. Coba buka Dasbor di sesi browser baru.

Menonaktifkan otentikasi Amazon Cognito untuk Dasbor OpenSearch

Gunakan prosedur berikut untuk menonaktifkan otentikasi Amazon Cognito untuk Dasbor.

Untuk menonaktifkan otentikasi Amazon Cognito untuk Dasbor (konsol)

1. Buka konsol OpenSearch Layanan Amazon di https://console.aws.amazon.com/aos/home/.
2. Di bawah Domain, pilih domain yang ingin Anda konfigurasikan.
3. Pilih Tindakan, Edit konfigurasi keamanan.
4. Hapus pilihan Aktifkan otentikasi Amazon Cognito.
5. Pilih Save changes (Simpan perubahan).

> **Important**
>
> Jika Anda tidak lagi membutuhkan kolam pengguna Amazon Cognito dan kolam identitas, hapus kolam tersebut. Jika tidak, Anda terus dikenakan biaya.

Menghapus domain yang menggunakan autentikasi Amazon Cognito untuk Dasbor OpenSearch

Untuk mencegah domain yang menggunakan autentikasi Amazon Cognito untuk Dasbor macet dalam status konfigurasi Pemrosesan, OpenSearch hapus domain Layanan sebelum menghapus kumpulan pengguna dan identitas Amazon Cognito terkait.

Menggunakan peran terkait layanan untuk Amazon Service OpenSearch

OpenSearch Layanan Amazon menggunakan peran AWS Identity and Access Management terkait layanan (IAM). Peran terkait layanan adalah jenis peran IAM unik yang ditautkan langsung ke Layanan. OpenSearch Peran terkait layanan telah ditentukan sebelumnya oleh OpenSearch Layanan dan mencakup semua izin yang diperlukan layanan untuk memanggil AWS layanan lain atas nama Anda.

Peran terkait layanan membuat pengaturan OpenSearch Layanan lebih mudah karena Anda tidak perlu menambahkan izin yang diperlukan secara manual. OpenSearch Layanan mendefinisikan

izin peran terkait layanan, dan kecuali ditentukan lain, hanya OpenSearch Layanan yang dapat mengambil perannya. Izin yang ditentukan mencakup kebijakan kepercayaan dan kebijakan izin, serta bahwa kebijakan izin tidak dapat dilampirkan ke entitas IAM lainnya. Untuk pembaruan kebijakan peran dan izin terkait layanan, lihat Riwayat dokumen untuk Amazon Service. OpenSearch

Untuk informasi tentang layanan lain yang mendukung peran terkait layanan, lihat Layanan AWS yang Bekerja bersama IAM dan mencari layanan yang memiliki Ya dalam Peran Terkait Layanan. Pilih Ya dengan tautan untuk melihat dokumentasi peran terkait layanan untuk layanan tersebut.

Topik

- Menggunakan peran terkait layanan untuk membuat domain VPC
- Menggunakan peran terkait layanan untuk membuat OpenSearch koleksi Tanpa Server
- Menggunakan peran terkait layanan untuk membuat OpenSearch saluran pipa Ingestion

Menggunakan peran terkait layanan untuk membuat domain VPC

OpenSearch Layanan Amazon menggunakan peran AWS Identity and Access Management terkait layanan (IAM). Peran terkait layanan adalah jenis peran IAM unik yang ditautkan langsung ke Layanan. OpenSearch Peran terkait layanan telah ditentukan sebelumnya oleh OpenSearch Layanan dan mencakup semua izin yang diperlukan layanan untuk memanggil AWS layanan lain atas nama Anda.

OpenSearch Layanan menggunakan peran terkait layanan bernama AWSServiceRoleForAmazonOpenSearchService, yang memberikan izin minimum Amazon EC2 dan Elastic Load Balancing yang diperlukan untuk peran tersebut guna mengaktifkan akses VPC untuk domain.

Peran Legacy Elasticsearch

Amazon OpenSearch Service menggunakan peran terkait layanan yang disebut. `AWSServiceRoleForAmazonOpenSearchService` Akun Anda mungkin juga berisi peran terkait layanan lama yang disebut`AWSServiceRoleForAmazonElasticsearchService`, yang berfungsi dengan titik akhir API Elasticsearch yang tidak digunakan lagi.

Jika peran Elasticsearch lama tidak ada di akun Anda, OpenSearch Layanan akan secara otomatis membuat peran OpenSearch terkait layanan baru saat pertama kali Anda membuat domain. OpenSearch Jika tidak, akun Anda akan terus menggunakan peran Elasticsearch. Agar pembuatan

otomatis ini berhasil, Anda harus memiliki izin untuk `iam:CreateServiceLinkedRole` tindakan tersebut.

Izin

`AWSServiceRoleForAmazonOpenSearchService` peran terkait layanan memercayakan layanan berikut untuk menjalankan peran tersebut:

- `opensearchservice.amazonaws.com`

Kebijakan izin peran bernama `AmazonOpenSearchServiceRolePolicy`memungkinkan OpenSearch Layanan untuk menyelesaikan tindakan berikut pada sumber daya yang ditentukan:

- Tindakan: `acm:DescribeCertificate` pada *
- Tindakan: `cloudwatch:PutMetricData` pada *
- Tindakan: `ec2:CreateNetworkInterface` pada *
- Tindakan: `ec2:DeleteNetworkInterface` pada *
- Tindakan: `ec2:DescribeNetworkInterfaces` pada *
- Tindakan: `ec2:ModifyNetworkInterfaceAttribute` pada *
- Tindakan: `ec2:DescribeSecurityGroups` pada *
- Tindakan: `ec2:DescribeSubnets` pada *
- Tindakan: `ec2:DescribeVpcs` pada *
- Tindakan: `ec2:CreateTags` pada semua antarmuka jaringan dan titik akhir VPC
- Tindakan: `ec2:DescribeTags` pada *
- Tindakan: `ec2:CreateVpcEndpoint` pada semua VPC, grup keamanan, subnet, dan tabel rute, serta semua titik akhir VPC saat permintaan berisi tag `OpenSearchManaged=true`
- Tindakan: `ec2:ModifyVpcEndpoint` pada semua VPC, grup keamanan, subnet, dan tabel rute, serta semua titik akhir VPC saat permintaan berisi tag `OpenSearchManaged=true`
- Tindakan: `ec2:DeleteVpcEndpoints` pada semua titik akhir saat permintaan berisi tag `OpenSearchManaged=true`
- Tindakan: `ec2:AssignIpv6Addresses` pada *
- Tindakan: `ec2:UnAssignIpv6Addresses` pada *
- Tindakan: `elasticloadbalancing:AddListenerCertificates` pada *
- Tindakan: `elasticloadbalancing:RemoveListenerCertificates` pada *

Anda harus mengonfigurasikan izin untuk mengizinkan entitas IAM (seperti pengguna, grup, atau peran) untuk membuat, menyunting, atau menghapus peran terhubung dengan layanan. Untuk informasi selengkapnya, lihat Izin peran tertaut layanan dalam Panduan Pengguna IAM.

Membuat peran terkait layanan

Anda tidak perlu membuat peran terkait layanan secara manual. Saat Anda membuat domain berkemampuan VPC menggunakanAWS Management Console, OpenSearch Layanan membuat peran terkait layanan untuk Anda. Agar pembuatan otomatis ini berhasil, Anda harus memiliki izin untuk `iam:CreateServiceLinkedRole` tindakan tersebut.

Anda juga dapat menggunakan konsol IAM, CLI IAM, atau API IAM untuk membuat peran tertaut layanan secara manual. Untuk informasi selengkapnya, lihat Membuat peran tertaut layanan dalam Panduan Pengguna IAM.

Mengedit peran terkait layanan

OpenSearch Layanan tidak mengizinkan Anda mengedit peran `AWSServiceRoleForAmazonOpenSearchService` terkait layanan. Setelah membuat peran terkait layanan, Anda tidak dapat mengubah nama peran karena berbagai entitas mungkin mereferensikan peran tersebut. Namun, Anda dapat menyunting penjelasan peran menggunakan IAM. Untuk informasi selengkapnya, lihat Mengedit peran yang terkait dengan layanan dalam Panduan Pengguna IAM.

Menghapus peran terkait layanan

Jika Anda tidak perlu lagi menggunakan fitur atau layanan yang memerlukan peran terkait layanan, kami merekomendasikan Anda menghapus peran tersebut. Dengan begitu, Anda tidak memiliki entitas yang tidak digunakan dan tidak dipantau atau dipelihara secara aktif. Namun, Anda harus membersihkan peran tertaut layanan terlebih dahulu sebelum dapat menghapusnya secara manual.

Membersihkan peran terkait layanan

Sebelum Anda dapat menggunakan IAM untuk menghapus peran terkait layanan, Anda harus mengonfirmasi terlebih dahulu bahwa peran tersebut tidak memiliki sesi aktif dan menghapus sumber daya yang digunakan oleh peran tersebut.

Untuk memastikan peran tertaut layanan memiliki sesi aktif di konsol IAM

1. Masuk ke AWS Management Console dan buka konsol IAM di https://console.aws.amazon.com/iam/.

2. Di panel navigasi konsol IAM, pilih Peran. Lalu pilih nama (bukan kotak centang) dari peran `AWSServiceRoleForAmazonOpenSearchService`.
3. Pada halaman Ringkasan untuk peran yang dipilih, pilih tab Penasihat Akses.
4. Pada tab Penasihat Akses, tinjau aktivitas terbaru untuk peran tertaut layanan itu.

> **Note**
>
> Jika tidak yakin apakah OpenSearch Layanan menggunakan `AWSServiceRoleForAmazonOpenSearchService` peran tersebut, Anda dapat mencoba menghapus peran tersebut. Jika layanan menggunakan peran, maka penghapusan gagal dan Anda dapat melihat sumber daya menggunakan peran tersebut. Jika peran sedang digunakan, maka Anda harus menunggu sesi berakhir sebelum Anda dapat menghapus peran, dan/atau menghapus sumber daya menggunakan peran tersebut. Anda tidak dapat mencabut sesi untuk peran terkait layanan.

Menghapus peran tertaut layanan secara manual

Hapus peran terkait layanan dari konsol IAM, API, atau CLI. AWS Untuk petunjuknya, lihat Menghapus peran terkait layanan di Panduan Pengguna IAM.

Menggunakan peran terkait layanan untuk membuat OpenSearch koleksi Tanpa Server

OpenSearch Serverless menggunakan peran terkait AWS Identity and Access Management layanan (IAM). Peran terkait layanan adalah jenis peran IAM unik yang ditautkan langsung ke Layanan. OpenSearch Peran terkait layanan telah ditentukan sebelumnya oleh OpenSearch Layanan dan mencakup semua izin yang diperlukan layanan untuk memanggil AWS layanan lain atas nama Anda.

OpenSearch Tanpa server menggunakan nama peran terkait layanan AWSServiceRoleForAmazonOpenSearchServerless, yang memberikan izin yang diperlukan agar peran tersebut memublikasikan metrik terkait tanpa server CloudWatch ke akun Anda.

Izin peran terkait layanan untuk Tanpa Server OpenSearch

OpenSearch Tanpa server menggunakan peran terkait layanan bernama AWSServiceRoleForAmazonOpenSearchServerless, yang memungkinkan OpenSearch Tanpa Server memanggil layanan atas nama Anda. AWS

Peran AWSServiceRoleForAmazonOpenSearchServerless terkait layanan mempercayai layanan berikut untuk mengambil peran:

- `observability.aoss.amazonaws.com`

Kebijakan izin peran bernama `AmazonOpenSearchServerlessServiceRolePolicy` memungkinkan OpenSearch Tanpa Server untuk menyelesaikan tindakan berikut pada sumber daya yang ditentukan:

- Tindakan: `cloudwatch:PutMetricData` pada semua AWS sumber daya

> **Note**
>
> Kebijakan ini menyertakan kunci kondisi`{"StringEquals": {"cloudwatch:namespace": "AWS/AOSS"}}`, yang berarti bahwa peran yang ditautkan layanan hanya dapat mengirim data metrik ke namespace. `AWS/AOSS` CloudWatch

Anda harus mengonfigurasi izin untuk mengizinkan entitas IAM (seperti pengguna, grup, atau peran) untuk membuat, mengedit, atau menghapus peran terkait layanan. Untuk informasi selengkapnya, lihat Izin peran tertaut layanan dalam Panduan Pengguna IAM.

Membuat peran terkait layanan untuk Tanpa Server OpenSearch

Anda tidak perlu membuat peran terkait layanan secara manual. Saat Anda membuat koleksi OpenSearch Tanpa Server di, APIAWS Management Console, atau AWS APIAWS CLI, OpenSearch Tanpa Server akan membuat peran terkait layanan untuk Anda.

> **Note**
>
> Pertama kali Anda membuat koleksi, Anda harus ditetapkan `iam:CreateServiceLinkedRole` dalam kebijakan berbasis identitas.

Jika Anda menghapus peran tertaut layanan ini, dan ingin membuatnya lagi, Anda dapat mengulangi proses yang sama untuk membuat kembali peran tersebut di akun Anda. Saat Anda membuat koleksi OpenSearch Tanpa Server, OpenSearch Tanpa Server akan membuat peran terkait layanan untuk Anda lagi.

Anda juga dapat menggunakan konsol IAM untuk membuat peran terkait layanan dengan kasus penggunaan Amazon Tanpa OpenSearch Server. Di AWS CLI atau AWS API, buat peran terkait layanan dengan nama `observability.aoss.amazonaws.com` layanan:

```
aws iam create-service-linked-role --aws-service-name
 "observability.aoss.amazonaws.com"
```

Untuk informasi selengkapnya, lihat Membuat peran tertaut layanan dalam Panduan Pengguna IAM. Jika Anda menghapus peran tertaut layanan ini, Anda dapat mengulang proses yang sama untuk membuat peran tersebut lagi.

Mengedit peran terkait layanan untuk Tanpa Server OpenSearch

OpenSearch Tanpa server tidak memungkinkan Anda untuk mengedit peran terkait AWSServiceRoleForAmazonOpenSearchServerless layanan. Setelah membuat peran terkait layanan, Anda tidak dapat mengubah nama peran karena berbagai entitas mungkin mereferensikan peran tersebut. Namun, Anda dapat menyunting penjelasan peran menggunakan IAM. Untuk informasi selengkapnya, lihat Mengedit peran yang terkait dengan layanan dalam Panduan Pengguna IAM.

Menghapus peran terkait layanan untuk Tanpa Server OpenSearch

Jika Anda tidak perlu lagi menggunakan fitur atau layanan yang memerlukan peran terkait layanan, kami merekomendasikan Anda menghapus peran tersebut. Ini mencegah Anda memiliki entitas yang tidak terpakai yang tidak dipantau atau dipelihara secara aktif. Tetapi, Anda harus membersihkan sumber daya peran terkait layanan sebelum menghapusnya secara manual.

Untuk menghapus AWSServiceRoleForAmazonOpenSearchServerless, Anda harus terlebih dahulu menghapus semua koleksi OpenSearch Tanpa Server di. Akun AWS

> **Note**
>
> Jika OpenSearch Tanpa Server menggunakan peran saat Anda mencoba menghapus sumber daya, maka penghapusan mungkin gagal. Jika hal itu terjadi, tunggu beberapa menit dan coba mengoperasikannya lagi.

Untuk menghapus peran terkait layanan secara manual menggunakan IAM

Gunakan konsol IAM, AWS CLI, atau AWS API untuk menghapus peran terkait layanan AWSServiceRoleForAmazonOpenSearchServerless. Untuk informasi selengkapnya, lihat Menghapus peran tertaut layanan dalam Panduan Pengguna IAM.

Wilayah yang Didukung untuk OpenSearch peran terkait layanan Tanpa Server

OpenSearch Dukungan tanpa server menggunakan peran AWSServiceRoleForAmazonOpenSearchServerless terkait layanan di setiap Wilayah tempat OpenSearch Tanpa Server tersedia. Untuk daftar Wilayah yang didukung, lihat titik akhir dan kuota Amazon OpenSearch Tanpa Server di. Referensi Umum AWS

Menggunakan peran terkait layanan untuk membuat OpenSearch saluran pipa Ingestion

Amazon OpenSearch Ingestion menggunakan peran terkait AWS Identity and Access Management layanan (IAM). Peran terkait layanan adalah jenis unik peran IAM yang terkait langsung dengan Ingestion. OpenSearch Peran terkait layanan telah ditentukan sebelumnya oleh OpenSearch Ingestion dan mencakup semua izin yang diperlukan layanan untuk memanggil layanan lain atas nama Anda. AWS

OpenSearch Ingestion menggunakan peran terkait layanan bernama. AWSServiceRoleForAmazonOpenSearchIngestion Kebijakan terlampir memberikan izin yang diperlukan untuk peran untuk membuat virtual private cloud (VPC) antara akun Anda OpenSearch dan Ingestion, dan untuk CloudWatch mempublikasikan metrik ke akun Anda.

Izin

`AWSServiceRoleForAmazonOpenSearchIngestion` peran terkait layanan memercayakan layanan berikut untuk menjalankan peran tersebut:

- `osis.amazon.com`

Kebijakan izin peran bernama `AmazonOpenSearchIngestionServiceRolePolicy` memungkinkan OpenSearch Ingestion untuk menyelesaikan tindakan berikut pada sumber daya yang ditentukan:

- Tindakan: `ec2:DescribeSubnets` pada *
- Tindakan: `ec2:DescribeSecurityGroups` pada *

- Tindakan: `ec2:DeleteVpcEndpoints` pada *
- Tindakan: `ec2:CreateVpcEndpoint` pada *
- Tindakan: `ec2:DescribeVpcEndpoints` pada *
- Tindakan: `ec2:CreateTags` pada `arn:aws:ec2:*:*:network-interface/*`
- Tindakan: `cloudwatch:PutMetricData` pada `cloudwatch:namespace": "AWS/OSIS"`

Anda harus mengonfigurasikan izin untuk mengizinkan entitas IAM (seperti pengguna, grup, atau peran) untuk membuat, menyunting, atau menghapus peran terhubung dengan layanan. Untuk informasi selengkapnya, lihat Izin peran tertaut layanan dalam Panduan Pengguna IAM.

Membuat peran terkait layanan untuk Ingestion OpenSearch

Anda tidak perlu membuat peran terkait layanan secara manual. Saat Anda membuat pipeline OpenSearch Ingestion diAWS Management Console, the, atau AWS APIAWS CLI, OpenSearch Ingestion akan membuat peran terkait layanan untuk Anda.

Jika Anda menghapus peran tertaut layanan ini, dan ingin membuatnya lagi, Anda dapat mengulangi proses yang sama untuk membuat kembali peran tersebut di akun Anda. Saat Anda membuat pipeline OpenSearch Ingestion, OpenSearch Ingestion membuat peran terkait layanan untuk Anda lagi.

Mengedit peran terkait layanan untuk Ingestion OpenSearch

OpenSearch Ingestion tidak memungkinkan Anda untuk mengedit peran terkait `AWSServiceRoleForAmazonOpenSearchIngestion` layanan. Setelah Anda membuat peran terkait layanan, Anda tidak dapat mengubah nama peran karena berbagai entitas mungkin mereferensikan peran tersebut. Namun, Anda dapat mengedit penjelasan peran menggunakan IAM. Untuk informasi selengkapnya, lihat Mengedit peran yang terkait dengan layanan dalam Panduan Pengguna IAM.

Menghapus peran terkait layanan untuk Ingestion OpenSearch

Jika Anda tidak perlu lagi menggunakan fitur atau layanan yang memerlukan peran terkait layanan, kami menyarankan 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.

Membersihkan peran yang terhubung dengan layanan

Sebelum dapat menggunakan IAM untuk menghapus peran tertaut-layanan, Anda harus terlebih dahulu menghapus semua sumber daya yang digunakan oleh peran tersebut.

> **Note**
>
> Jika OpenSearch Ingestion menggunakan peran saat Anda mencoba menghapus sumber daya, maka penghapusan mungkin gagal. Jika hal itu terjadi, tunggu beberapa menit dan coba mengoperasikannya lagi.

Untuk menghapus sumber daya OpenSearch Ingestion yang digunakan oleh **`AWSServiceRoleForAmazonOpenSearchIngestion`**

1. Arahkan ke konsol Amazon OpenSearch Service dan pilih Ingestion.
2. Hapus semua saluran pipa. Untuk petunjuk, lihat the section called “Menghapus Alur”.

Menghapus peran terkait layanan untuk Ingestion OpenSearch

Anda dapat menggunakan konsol OpenSearch Ingestion untuk menghapus peran terkait layanan.

Untuk menghapus peran tertaut layanan (konsol)

1. Arahkan ke konsol IAM.
2. Pilih Peran dan cari AWSServiceRoleForAmazonOpenSearchIngestionperannya.
3. Pilih peran dan pilih Hapus.

Contoh kode untuk AmazonOpenSearchLayanan

Bab ini berisi kode sampel umum untuk bekerja dengan AmazonOpenSearchLayanan: Permintaan HTTP masuk dalam berbagai bahasa pemrograman, mengompresi badan permintaan HTTP, dan menggunakanAWSSDK untuk membuat domain.

Topik

Kompatibilitas klien Elasticsearch

Versi terbaru dari klien Elasticsearch mungkin menyertakan pemeriksaan lisensi atau versi yang secara artifitis merusak kompatibilitas. Tabel berikut mencakup rekomendasi seputar versi klien mana yang akan digunakan untuk kompatibilitas terbaikOpenSearchLayanan.

> **Important**
>
> Versi klien ini kedaluwarsa dan tidak diperbarui dengan dependensi terbaru, termasuk Log4j. Kami sangat merekomendasikan menggunakanOpenSearchversi klien bila memungkinkan.

Klien	Versi yang disarankan
Klien REST tingkat rendah Java	7.13.4
Klien REST tingkat tinggi Java	7.13.4
Klien Python Elasticsearch	7.13.4
Klien Ruby Elasticsearch	7.13.3
Klien Node.js Elasticsearch	7.13.0

Mengompresi permintaan HTTP di Amazon OpenSearch Service

Anda dapat mengompres permintaan dan respons HTTP di domain Amazon OpenSearch Service menggunakan kompresi gzip. Kompresi Gzip dapat membantu Anda mengurangi ukuran dokumen dan menurunkan penggunaan bandwidth dan latensi, sehingga meningkatkan kecepatan transfer.

Kompresi Gzip didukung untuk semua domain yang menjalankan OpenSearch atau Elasticsearch 6.0 atau yang lebih baru. Beberapa klien OpenSearch memiliki dukungan bawaan untuk kompresi gzip, dan banyak bahasa pemrograman memiliki pustaka yang menyederhanakan prosesnya.

Mengaktifkan kompresi gzip

Jangan bingung dengan pengaturan OpenSearch yang serupa,`http_compression.enabled`khusus untuk OpenSearch Service dan memungkinkan atau menonaktifkan kompresi gzip pada domain. Domain yang menjalankan OpenSearch atau Elasticsearch 7.xmemiliki kompresi gzip diaktifkan secara default, sedangkan domain yang menjalankan Elasticsearch 6.xmemilikinya dinonaktifkan secara default.

Untuk mengaktifkan kompresi gzip, kirim permintaan berikut:

```
PUT _cluster/settings
{
  "persistent" : {
    "http_compression.enabled": true
  }
}
```

Permintaan ke `_cluster/settings` harus tidak dikompresi, jadi Anda mungkin perlu menggunakan klien terpisah atau permintaan HTTP standar untuk memperbarui pengaturan klaster.

Header yang dibutuhkan

Ketika menyertakan isi permintaan terkompresi gzip, pertahankan standar header `Content-Type: application/json`, dan tambahkan header `Content-Encoding: gzip`. Untuk menerima respon terkompresi gzip, tambahkan header `Accept-Encoding: gzip` juga. Jika klien OpenSearch mendukung kompresi gzip, kemungkinan termasuk header ini secara otomatis.

Contoh kode (Python 3)

Contoh berikut menggunakanopensearch-pyuntuk melakukan kompresi dan mengirim permintaan. Kode ini menandatangani permintaan menggunakan kredensial IAM Anda.

```
from opensearchpy import OpenSearch, RequestsHttpConnection
from requests_aws4auth import AWS4Auth
import boto3

host = '' # e.g. my-test-domain.us-east-1.es.amazonaws.com
region = '' # e.g. us-west-1
service = 'es'
credentials = boto3.Session().get_credentials()
awsauth = AWS4Auth(credentials.access_key, credentials.secret_key, region, service,
 session_token=credentials.token)

# Create the client.
search = OpenSearch(
    hosts = [{'host': host, 'port': 443}],
    http_auth = awsauth,
    use_ssl = True,
    verify_certs = True,
    http_compress = True, # enables gzip compression for request bodies
    connection_class = RequestsHttpConnection
)

document = {
  "title": "Moneyball",
  "director": "Bennett Miller",
  "year": "2011"
}

# Send the request.
print(search.index(index='movies', id='1', body=document, refresh=True))

# print(search.index(index='movies', doc_type='_doc', id='1', body=document,
 refresh=True))
```

Sebagai alternatif, Anda dapat menentukan tajuk yang tepat, mengompres sendiri isi permintaan, dan menggunakan pustaka HTTP standar seperti Permintaan. Kode ini menandatangani permintaan menggunakan kredensial dasar HTTP, yang mungkin didukung domain Anda jika Anda menggunakan kontrol akses detail.

```
import requests
import gzip
import json

base_url = '' # The domain with https:// and a trailing slash. For example, https://my-
test-domain.us-east-1.es.amazonaws.com/
auth = ('master-user', 'master-user-password') # For testing only. Don't store
 credentials in code.

headers = {'Accept-Encoding': 'gzip', 'Content-Type': 'application/json',
           'Content-Encoding': 'gzip'}

document = {
  "title": "Moneyball",
  "director": "Bennett Miller",
  "year": "2011"
}

# Compress the document.
compressed_document = gzip.compress(json.dumps(document).encode())

# Send the request.
path = 'movies/_doc?refresh=true'
url = base_url + path
response = requests.post(url, auth=auth, headers=headers, data=compressed_document)
print(response.status_code)
print(response.text)
```

MenggunakanAWSSDK untuk berinteraksi dengan AmazonOpenSearchLayanan

Bagian ini mencakup contoh cara menggunakanAWSSDK untuk berinteraksi dengan AmazonOpenSearchAPI konfigurasi layanan. Contoh kode ini menunjukkan cara membuat, memperbarui, dan menghapusOpenSearchDomain layanan.

Java

Bagian ini mencakup contoh untuk versi 1 dan 2AWS SDK for Java.

Version 2

Contoh ini menggunakanOpenSearchClientBuilderkonstruktor dari versi 2 dariAWS SDK for Javauntuk membuatOpenSearchdomain, perbarui konfigurasinya, dan hapus. Batalkan komentar panggilan ke `waitForDomainProcessing` (dan komentari panggilan ke `deleteDomain`) untuk memungkinkan domain online dan dapat digunakan.

```
package com.example.samples;

import java.util.concurrent.TimeUnit;
import software.amazon.awssdk.regions.Region;
import software.amazon.awssdk.services.opensearch.OpenSearchClient;
import software.amazon.awssdk.services.opensearch.model.ClusterConfig;
import software.amazon.awssdk.services.opensearch.model.EBSOptions;
import software.amazon.awssdk.services.opensearch.model.CognitoOptions;
import software.amazon.awssdk.services.opensearch.model.NodeToNodeEncryptionOptions;
import software.amazon.awssdk.services.opensearch.model.CreateDomainRequest;
import software.amazon.awssdk.services.opensearch.model.CreateDomainResponse;
import software.amazon.awssdk.services.opensearch.model.DescribeDomainRequest;
import software.amazon.awssdk.services.opensearch.model.UpdateDomainConfigRequest;
import software.amazon.awssdk.services.opensearch.model.UpdateDomainConfigResponse;
import software.amazon.awssdk.services.opensearch.model.DescribeDomainResponse;
import software.amazon.awssdk.services.opensearch.model.DeleteDomainRequest;
import software.amazon.awssdk.services.opensearch.model.DeleteDomainResponse;
import software.amazon.awssdk.services.opensearch.model.OpenSearchException;
import software.amazon.awssdk.auth.credentials.DefaultCredentialsProvider;

/**
 * Sample class demonstrating how to use the Amazon Web Services SDK for Java to
 create, update,
 * and delete Amazon OpenSearch Service domains.
 */

public class OpenSearchSample {

    public static void main(String[] args) {

     String domainName = "my-test-domain";

     // Build the client using the default credentials chain.
        // You can use the CLI and run `aws configure` to set access key, secret
        // key, and default region.
```

```
        OpenSearchClient client = OpenSearchClient.builder()
          // Unnecessary, but lets you use a region different than your default.
          .region(Region.US_EAST_1)
          // Unnecessary, but if desired, you can use a different provider chain.
          .credentialsProvider(DefaultCredentialsProvider.create())
                    .build();

        // Create a new domain, update its configuration, and delete it.
        createDomain(client, domainName);
        //waitForDomainProcessing(client, domainName);
        updateDomain(client, domainName);
        //waitForDomainProcessing(client, domainName);
        deleteDomain(client, domainName);
    }

    /**
     * Creates an Amazon OpenSearch Service domain with the specified options.
     * Some options require other Amazon Web Services resources, such as an Amazon
Cognito user pool
     * and identity pool, whereas others require just an instance type or instance
     * count.
     *
     * @param client
     *            The client to use for the requests to Amazon OpenSearch Service
     * @param domainName
     *            The name of the domain you want to create
     */

    public static void createDomain(OpenSearchClient client, String domainName) {

     // Create the request and set the desired configuration options

        try {

            ClusterConfig clusterConfig = ClusterConfig.builder()
                    .dedicatedMasterEnabled(true)
                    .dedicatedMasterCount(3)
                    // Small, inexpensive instance types for testing. Not
recommended for production.
                    .dedicatedMasterType("t2.small.search")
                    .instanceType("t2.small.search")
                    .instanceCount(5)
                    .build();
```

```
            // Many instance types require EBS storage.
            EBSOptions ebsOptions = EBSOptions.builder()
                    .ebsEnabled(true)
                    .volumeSize(10)
                    .volumeType("gp2")
                    .build();

            NodeToNodeEncryptionOptions encryptionOptions =
 NodeToNodeEncryptionOptions.builder()
                    .enabled(true)
                    .build();

            CreateDomainRequest createRequest = CreateDomainRequest.builder()
                    .domainName(domainName)
                    .engineVersion("OpenSearch_1.0")
                    .clusterConfig(clusterConfig)
                    .ebsOptions(ebsOptions)
                    .nodeToNodeEncryptionOptions(encryptionOptions)
                    // You can uncomment this line and add your account ID, a
 username, and the
                    // domain name to add an access policy.
                    // .accessPolicies("{\"Version\":\"2012-10-17\",
\"Statement\":[{\"Effect\":\"Allow\",\"Principal\":{\"AWS\":
[\"arn:aws:iam::123456789012:user/user-name\"]},\"Action\":[\"es:*\"],\"Resource\":
\"arn:aws:es:region:123456789012:domain/domain-name/*\"}]}")
                    .build();

            // Make the request.
            System.out.println("Sending domain creation request...");
            CreateDomainResponse createResponse =
 client.createDomain(createRequest);
            System.out.println("Domain status:
 "+createResponse.domainStatus().toString());
            System.out.println("Domain ID:
 "+createResponse.domainStatus().domainId());


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

    /**
```

```
    * Updates the configuration of an Amazon OpenSearch Service domain with the
    * specified options. Some options require other Amazon Web Services resources,
such as an
    * Amazon Cognito user pool and identity pool, whereas others require just an
    * instance type or instance count.
    *
    * @param client
    *            The client to use for the requests to Amazon OpenSearch Service
    * @param domainName
    *            The name of the domain to update
    */

   public static void updateDomain(OpenSearchClient client, String domainName) {

    // Updates the domain to use three data instances instead of five.
       // You can uncomment the Cognito line and fill in the strings to enable
Cognito
       // authentication for OpenSearch Dashboards.

       try {

           ClusterConfig clusterConfig = ClusterConfig.builder()
                   .instanceCount(5)
                   .build();

           CognitoOptions cognitoOptions = CognitoOptions.builder()
                   .enabled(true)
                   .userPoolId("user-pool-id")
                   .identityPoolId("identity-pool-id")
                   .roleArn("role-arn")
                   .build();

           UpdateDomainConfigRequest updateRequest =
UpdateDomainConfigRequest.builder()
                   .domainName(domainName)
                   .clusterConfig(clusterConfig)
                   //.cognitoOptions(cognitoOptions)
                   .build();

           System.out.println("Sending domain update request...");
           UpdateDomainConfigResponse updateResponse =
client.updateDomainConfig(updateRequest);
           System.out.println("Domain config:
"+updateResponse.domainConfig().toString());
```

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

    /**
     * Deletes an Amazon OpenSearch Service domain. Deleting a domain can take
     * several minutes.
     *
     * @param client
     *             The client to use for the requests to Amazon OpenSearch Service
     * @param domainName
     *             The name of the domain that you want to delete
     */

    public static void deleteDomain(OpenSearchClient client, String domainName) {

        try {

            DeleteDomainRequest deleteRequest = DeleteDomainRequest.builder()
                    .domainName(domainName)
                    .build();

            System.out.println("Sending domain deletion request...");
            DeleteDomainResponse deleteResponse =
client.deleteDomain(deleteRequest);
            System.out.println("Domain status: "+deleteResponse.toString());


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

    /**
     * Waits for the domain to finish processing changes. New domains typically take
15-30 minutes
     * to initialize, but can take longer depending on the configuration. Most
updates to existing domains
```

```
     * take a similar amount of time. This method checks every 15 seconds and
 finishes only when
     * the domain's processing status changes to false.
     *
     * @param client
     *            The client to use for the requests to Amazon OpenSearch Service
     * @param domainName
     *            The name of the domain that you want to check
     */

    public static void waitForDomainProcessing(OpenSearchClient client, String
 domainName) {
        // Create a new request to check the domain status.
        DescribeDomainRequest describeRequest = DescribeDomainRequest.builder()
                .domainName(domainName)
                .build();

        // Every 15 seconds, check whether the domain is processing.
        DescribeDomainResponse describeResponse =
 client.describeDomain(describeRequest);
        while (describeResponse.domainStatus().processing()) {
            try {
                System.out.println("Domain still processing...");
                TimeUnit.SECONDS.sleep(15);
                describeResponse = client.describeDomain(describeRequest);
            } catch (InterruptedException e) {
                e.printStackTrace();
            }
        }

        // Once we exit that loop, the domain is available
        System.out.println("Amazon OpenSearch Service has finished processing
 changes for your domain.");
        System.out.println("Domain description: "+describeResponse.toString());
    }
}
```

Version 1

Contoh ini menggunakanAWSElasticsearchClientBuilderkonstruktor dari versi 1AWS SDK for Javauntuk membuat domain Elasticsearch lama, perbarui konfigurasinya, dan hapus. Batalkan komentar panggilan ke `waitForDomainProcessing` (dan komentari panggilan ke `deleteDomain`) untuk memungkinkan domain online dan dapat digunakan.

```
package com.amazonaws.samples;

import java.util.concurrent.TimeUnit;
import com.amazonaws.auth.DefaultAWSCredentialsProviderChain;
import com.amazonaws.regions.Regions;
import com.amazonaws.services.elasticsearch.AWSElasticsearch;
import com.amazonaws.services.elasticsearch.AWSElasticsearchClientBuilder;
import com.amazonaws.services.elasticsearch.model.CreateElasticsearchDomainRequest;
import com.amazonaws.services.elasticsearch.model.CreateElasticsearchDomainResult;
import com.amazonaws.services.elasticsearch.model.DeleteElasticsearchDomainRequest;
import com.amazonaws.services.elasticsearch.model.DeleteElasticsearchDomainResult;
import
 com.amazonaws.services.elasticsearch.model.DescribeElasticsearchDomainRequest;
import com.amazonaws.services.elasticsearch.model.DescribeElasticsearchDomainResult;
import com.amazonaws.services.elasticsearch.model.EBSOptions;
import com.amazonaws.services.elasticsearch.model.ElasticsearchClusterConfig;
import com.amazonaws.services.elasticsearch.model.ResourceNotFoundException;
import
 com.amazonaws.services.elasticsearch.model.UpdateElasticsearchDomainConfigRequest;
import
 com.amazonaws.services.elasticsearch.model.UpdateElasticsearchDomainConfigResult;
import com.amazonaws.services.elasticsearch.model.VolumeType;

/**
 * Sample class demonstrating how to use the Amazon Web Services SDK for Java to
 create, update,
 * and delete Amazon OpenSearch Service domains.
 */

public class OpenSearchSample {

    public static void main(String[] args) {

        final String domainName = "my-test-domain";

        // Build the client using the default credentials chain.
        // You can use the CLI and run `aws configure` to set access key, secret
        // key, and default region.
        final AWSElasticsearch client = AWSElasticsearchClientBuilder
                .standard()
                // Unnecessary, but lets you use a region different than your
 default.
                .withRegion(Regions.US_WEST_2)
```

```
                // Unnecessary, but if desired, you can use a different provider
chain.
                .withCredentials(new DefaultAWSCredentialsProviderChain())
                .build();

        // Create a new domain, update its configuration, and delete it.
        createDomain(client, domainName);
        // waitForDomainProcessing(client, domainName);
        updateDomain(client, domainName);
        // waitForDomainProcessing(client, domainName);
        deleteDomain(client, domainName);
    }

    /**
     * Creates an Amazon OpenSearch Service domain with the specified options.
     * Some options require other Amazon Web Services resources, such as an Amazon
Cognito user pool
     * and identity pool, whereas others require just an instance type or instance
     * count.
     *
     * @param client
     *            The client to use for the requests to Amazon OpenSearch Service
     * @param domainName
     *            The name of the domain you want to create
     */
    private static void createDomain(final AWSElasticsearch client, final String
domainName) {

        // Create the request and set the desired configuration options
        CreateElasticsearchDomainRequest createRequest = new
CreateElasticsearchDomainRequest()
                .withDomainName(domainName)
                .withElasticsearchVersion("7.10")
                .withElasticsearchClusterConfig(new ElasticsearchClusterConfig()
                        .withDedicatedMasterEnabled(true)
                        .withDedicatedMasterCount(3)
                        // Small, inexpensive instance types for testing. Not
recommended for production
                        // domains.
                        .withDedicatedMasterType("t2.small.elasticsearch")
                        .withInstanceType("t2.small.elasticsearch")
                        .withInstanceCount(5))
                // Many instance types require EBS storage.
                .withEBSOptions(new EBSOptions()
```

```
                        .withEBSEnabled(true)
                        .withVolumeSize(10)
                        .withVolumeType(VolumeType.Gp2));
                // You can uncomment this line and add your account ID, a username,
 and the
                // domain name to add an access policy.
                // .withAccessPolicies("{\"Version\":\"2012-10-17\",
\"Statement\":[{\"Effect\":\"Allow\",\"Principal\":{\"AWS\":
[\"arn:aws:iam::123456789012:user/user-name\"]},\"Action\":[\"es:*\"],\"Resource\":
\"arn:aws:es:region:123456789012:domain/domain-name/*\"}]}")


        // Make the request.
        System.out.println("Sending domain creation request...");
        CreateElasticsearchDomainResult createResponse =
 client.createElasticsearchDomain(createRequest);
        System.out.println("Domain creation response from Amazon OpenSearch
 Service:");
        System.out.println(createResponse.getDomainStatus().toString());
    }

    /**
     * Updates the configuration of an Amazon OpenSearch Service domain with the
     * specified options. Some options require other Amazon Web Services resources,
 such as an
     * Amazon Cognito user pool and identity pool, whereas others require just an
     * instance type or instance count.
     *
     * @param client
     *            The client to use for the requests to Amazon OpenSearch Service
     * @param domainName
     *            The name of the domain to update
     */
    private static void updateDomain(final AWSElasticsearch client, final String
 domainName) {
        try {
            // Updates the domain to use three data instances instead of five.
            // You can uncomment the Cognito lines and fill in the strings to enable
 Cognito
            // authentication for OpenSearch Dashboards.
            final UpdateElasticsearchDomainConfigRequest updateRequest = new
 UpdateElasticsearchDomainConfigRequest()
                    .withDomainName(domainName)
                    // .withCognitoOptions(new CognitoOptions()
```

```
                            // .withEnabled(true)
                            // .withUserPoolId("user-pool-id")
                            // .withIdentityPoolId("identity-pool-id")
                            // .withRoleArn("role-arn")
                    .withElasticsearchClusterConfig(new ElasticsearchClusterConfig()
                            .withInstanceCount(3));

            System.out.println("Sending domain update request...");
            final UpdateElasticsearchDomainConfigResult updateResponse = client
                    .updateElasticsearchDomainConfig(updateRequest);
            System.out.println("Domain update response from Amazon OpenSearch
Service:");
            System.out.println(updateResponse.toString());
        } catch (ResourceNotFoundException e) {
            System.out.println("Domain not found. Please check the domain name.");
        }
    }

    /**
     * Deletes an Amazon OpenSearch Service domain. Deleting a domain can take
     * several minutes.
     *
     * @param client
     *            The client to use for the requests to Amazon OpenSearch Service
     * @param domainName
     *            The name of the domain that you want to delete
     */
    private static void deleteDomain(final AWSElasticsearch client, final String
domainName) {
        try {
            final DeleteElasticsearchDomainRequest deleteRequest = new
DeleteElasticsearchDomainRequest()
                    .withDomainName(domainName);

            System.out.println("Sending domain deletion request...");
            final DeleteElasticsearchDomainResult deleteResponse =
client.deleteElasticsearchDomain(deleteRequest);
            System.out.println("Domain deletion response from Amazon OpenSearch
Service:");
            System.out.println(deleteResponse.toString());
        } catch (ResourceNotFoundException e) {
            System.out.println("Domain not found. Please check the domain name.");
        }
    }
```

```
   /**
    * Waits for the domain to finish processing changes. New domains typically take
15-30 minutes
    * to initialize, but can take longer depending on the configuration. Most
updates to existing domains
    * take a similar amount of time. This method checks every 15 seconds and
finishes only when
    * the domain's processing status changes to false.
    *
    * @param client
    *            The client to use for the requests to Amazon OpenSearch Service
    * @param domainName
    *            The name of the domain that you want to check
    */
   private static void waitForDomainProcessing(final AWSElasticsearch client, final
String domainName) {
       // Create a new request to check the domain status.
       final DescribeElasticsearchDomainRequest describeRequest = new
DescribeElasticsearchDomainRequest()
               .withDomainName(domainName);

       // Every 15 seconds, check whether the domain is processing.
       DescribeElasticsearchDomainResult describeResponse =
client.describeElasticsearchDomain(describeRequest);
       while (describeResponse.getDomainStatus().isProcessing()) {
           try {
               System.out.println("Domain still processing...");
               TimeUnit.SECONDS.sleep(15);
               describeResponse =
client.describeElasticsearchDomain(describeRequest);
           } catch (InterruptedException e) {
               e.printStackTrace();
           }
       }

       // Once we exit that loop, the domain is available
       System.out.println("Amazon OpenSearch Service has finished processing
changes for your domain.");
       System.out.println("Domain description response from Amazon OpenSearch
Service:");
       System.out.println(describeResponse.toString());
   }
```

```
}
```

Python

Contoh ini menggunakanOpenSearchServiceklien Python tingkat rendah dariAWS SDK for Python (Boto)untuk membuat domain, memperbarui konfigurasinya, dan menghapusnya.

```
import boto3
import botocore
from botocore.config import Config
import time

# Build the client using the default credential configuration.
# You can use the CLI and run 'aws configure' to set access key, secret
# key, and default region.

my_config = Config(
    # Optionally lets you specify a region other than your default.
    region_name='us-west-2'
)

client = boto3.client('opensearch', config=my_config)

domainName = 'my-test-domain'  # The name of the domain


def createDomain(client, domainName):
    """Creates an Amazon OpenSearch Service domain with the specified options."""
    response = client.create_domain(
        DomainName=domainName,
        EngineVersion='OpenSearch_1.0',
        ClusterConfig={
            'InstanceType': 't2.small.search',
            'InstanceCount': 5,
            'DedicatedMasterEnabled': True,
            'DedicatedMasterType': 't2.small.search',
            'DedicatedMasterCount': 3
        },
        # Many instance types require EBS storage.
        EBSOptions={
            'EBSEnabled': True,
            'VolumeType': 'gp2',
```

```
            'VolumeSize': 10
        },
        AccessPolicies="{\"Version\":\"2012-10-17\",\"Statement\":[{\"Effect\":\"Allow
\",\"Principal\":{\"AWS\":[\"arn:aws:iam::123456789012:user/user-name\"]},\"Action\":
[\"es:*\"],\"Resource\":\"arn:aws:es:us-west-2:123456789012:domain/my-test-domain/*
\"}]}",
        NodeToNodeEncryptionOptions={
            'Enabled': True
        }
    )
    print("Creating domain...")
    print(response)


def updateDomain(client, domainName):
    """Updates the domain to use three data nodes instead of five."""
    try:
        response = client.update_domain_config(
            DomainName=domainName,
            ClusterConfig={
                'InstanceCount': 3
            }
        )
        print('Sending domain update request...')
        print(response)

    except botocore.exceptions.ClientError as error:
        if error.response['Error']['Code'] == 'ResourceNotFoundException':
            print('Domain not found. Please check the domain name.')
        else:
            raise error


def deleteDomain(client, domainName):
    """Deletes an OpenSearch Service domain. Deleting a domain can take several
 minutes."""
    try:
        response = client.delete_domain(
            DomainName=domainName
        )
        print('Sending domain deletion request...')
        print(response)

    except botocore.exceptions.ClientError as error:
```

```
        if error.response['Error']['Code'] == 'ResourceNotFoundException':
            print('Domain not found. Please check the domain name.')
        else:
            raise error


def waitForDomainProcessing(client, domainName):
    """Waits for the domain to finish processing changes."""
    try:
        response = client.describe_domain(
            DomainName=domainName
        )
        # Every 15 seconds, check whether the domain is processing.
        while response["DomainStatus"]["Processing"] == True:
            print('Domain still processing...')
            time.sleep(15)
            response = client.describe_domain(
                DomainName=domainName)

        # Once we exit the loop, the domain is available.
        print('Amazon OpenSearch Service has finished processing changes for your
 domain.')
        print('Domain description:')
        print(response)

    except botocore.exceptions.ClientError as error:
        if error.response['Error']['Code'] == 'ResourceNotFoundException':
            print('Domain not found. Please check the domain name.')
        else:
            raise error


def main():
    """Create a new domain, update its configuration, and delete it."""
    createDomain(client, domainName)
    waitForDomainProcessing(client, domainName)
    updateDomain(client, domainName)
    waitForDomainProcessing(client, domainName)
    deleteDomain(client, domainName)
```

Node

Contoh ini menggunakan SDK versi 3 untukJavaScriptdi Node.jsOpenSearchklienuntuk membuat domain, memperbarui konfigurasinya, dan menghapusnya.

```
var {
    OpenSearchClient,
    CreateDomainCommand,
    DescribeDomainCommand,
    UpdateDomainConfigCommand,
    DeleteDomainCommand
} = require("@aws-sdk/client-opensearch");
var sleep = require('sleep');

var client = new OpenSearchClient();

var domainName = 'my-test-domain'

// Create a new domain, update its configuration, and delete it.
createDomain(client, domainName)
waitForDomainProcessing(client, domainName)
updateDomain(client, domainName)
waitForDomainProcessing(client, domainName)
deleteDomain(client, domainName)

async function createDomain(client, domainName) {
    // Creates an Amazon OpenSearch Service domain with the specified options.
    var command = new CreateDomainCommand({
        DomainName: domainName,
        EngineVersion: 'OpenSearch_1.0',
        ClusterConfig: {
        'InstanceType': 't2.small.search',
        'InstanceCount': 5,
        'DedicatedMasterEnabled': 'True',
        'DedicatedMasterType': 't2.small.search',
        'DedicatedMasterCount': 3
        },
        EBSOptions:{
            'EBSEnabled': 'True',
            'VolumeType': 'gp2',
            'VolumeSize': 10
        },
```

```
        AccessPolicies: "{\"Version\":\"2012-10-17\",\"Statement\":[{\"Effect\":\"Allow
\",\"Principal\":{\"AWS\":[\"arn:aws:iam::123456789012:user/user-name\"]},\"Action\":
[\"es:*\"],\"Resource\":\"arn:aws:es:us-east-1:123456789012:domain/my-test-domain/*
\"}]}",
         NodeToNodeEncryptionOptions:{
            'Enabled': 'True'
        }
    });
    const response = await client.send(command);
    console.log("Creating domain...");
    console.log(response);
}

async function updateDomain(client, domainName) {
    // Updates the domain to use three data nodes instead of five.
    var command = new UpdateDomainConfigCommand({
        DomainName: domainName,
        ClusterConfig: {
        'InstanceCount': 3
        }
    });
    const response = await client.send(command);
    console.log('Sending domain update request...');
    console.log(response);
}

async function deleteDomain(client, domainName) {
    // Deletes an OpenSearch Service domain. Deleting a domain can take several
 minutes.
    var command = new DeleteDomainCommand({
        DomainName: domainName
    });
    const response = await client.send(command);
    console.log('Sending domain deletion request...');
    console.log(response);
}

async function waitForDomainProcessing(client, domainName) {
    // Waits for the domain to finish processing changes.
    try {
        var command = new DescribeDomainCommand({
            DomainName: domainName
        });
        var response = await client.send(command);
```

```
        while (response.DomainStatus.Processing == true) {
            console.log('Domain still processing...')
            await sleep(15000) // Wait for 15 seconds, then check the status again
            function sleep(ms) {
                return new Promise((resolve) => {
                    setTimeout(resolve, ms);
                });
            }
            var response = await client.send(command);
        }
        // Once we exit the loop, the domain is available.
        console.log('Amazon OpenSearch Service has finished processing changes for your
 domain.');
        console.log('Domain description:');
        console.log(response);

    } catch (error) {
        if (error.name === 'ResourceNotFoundException') {
            console.log('Domain not found. Please check the domain name.');
            }
    };
}
```

Pengindeksan data di Amazon Service OpenSearch

Karena Amazon OpenSearch Service menggunakan REST API, ada banyak metode untuk mengindeks dokumen. Anda dapat menggunakan klien standar seperti curl atau bahasa pemrograman apa pun yang dapat mengirim permintaan HTTP. Untuk lebih menyederhanakan proses berinteraksi dengannya, OpenSearch Layanan memiliki klien untuk banyak bahasa pemrograman. Pengguna tingkat lanjut dapat langsung melompat kethe section called “Memuat data streaming ke OpenSearch Layanan”.

Kami sangat menyarankan agar Anda menggunakan Amazon OpenSearch Ingestion untuk menelan data, yang merupakan pengumpul data terkelola sepenuhnya yang dibangun dalam Layanan. OpenSearch Untuk informasi selengkapnya, lihat Amazon OpenSearch Ingestion.

Untuk pengenalan pengindeksan, lihat OpenSearchdokumentasi.

Pembatasan penamaan untuk indeks

OpenSearch Indeks layanan memiliki batasan penamaan berikut:

- Semua huruf harus huruf kecil.
- Nama indeks tidak dapat dimulai dengan _ atau -.
- Nama indeks tidak dapat berisi spasi, koma, :, ", *, +, /, \, |, ?, #, >, atau <.

Jangan sertakan informasi sensitif dalam indeks, jenis, atau nama ID dokumen. OpenSearch Layanan menggunakan nama-nama ini dalam Uniform Resource Identifiers (URI). Server dan aplikasi sering mencatat permintaan HTTP, yang dapat menyebabkan eksposur data yang tidak perlu jika URI berisi informasi sensitif:

```
2018-10-03T23:39:43 198.51.100.14 200 "GET https://opensearch-domain/dr-jane-doe/flu-
patients-2018/202-555-0100/ HTTP/1.1"
```

Bahkan jika Anda tidak memiliki Izin untuk melihat dokumen JSON terkait, Anda dapat menyimpulkan dari baris log palsu ini bahwa salah satu pasien Dr. Doe dengan nomor telepon 202-555-0100 terkena flu pada 2018.

Jika OpenSearch Layanan mendeteksi alamat IP asli atau percieved dalam nama indeks (misalnya,`my-index-12.34.56.78.91`), itu menutupi alamat IP. Panggilan untuk `_cat/indices` menghasilkan respons berikut:

```
green open my-index-x.x.x.x.91    soY19tBERoKo71WcEScidw 5 1 0 0   2kb  1kb
```

Untuk mencegah kebingungan yang tidak perlu, hindari memasukkan alamat IP dalam nama indeks.

Mengurangi ukuran respons

Tanggapan dari API `_index` dan `_bulk` mengandung sedikit informasi. Informasi ini dapat berguna untuk pemecahan masalah permintaan atau untuk menerapkan logika coba lagi, tetapi dapat menggunakan bandwidth yang cukup besar. Dalam contoh ini, pengindeksan dokumen 32 byte menghasilkan respons 339 byte (termasuk header):

```
PUT opensearch-domain/more-movies/_doc/1
{"title": "Back to the Future"}
```

Respon

```
{
  "_index": "more-movies",
  "_type": "_doc",
  "_id": "1",
  "_version": 4,
  "result": "updated",
  "_shards": {
    "total": 2,
    "successful": 2,
    "failed": 0
  },
  "_seq_no": 3,
  "_primary_term": 1
}
```

Ukuran respons ini mungkin tampak minimal, tetapi jika Anda mengindeks 1.000.000 dokumen per hari—sekitar 11,5 dokumen per detik—339 byte per respons menghasilkan lalu lintas unduhan 10,17 GB per bulan.

Jika biaya transfer data menjadi perhatian, gunakan `filter_path` parameter untuk mengurangi ukuran respons OpenSearch Layanan, tetapi berhati-hatilah untuk tidak menyaring bidang yang Anda butuhkan untuk mengidentifikasi atau mencoba kembali permintaan yang gagal. Bidang ini bervariasi oleh klien. `filter_path`Parameter berfungsi untuk semua API REST OpenSearch Layanan, tetapi sangat berguna dengan API yang sering Anda panggil, seperti `_bulk` API `_index` dan API:

```
PUT opensearch-domain/more-movies/_doc/1?filter_path=result,_shards.total
{"title": "Back to the Future"}
```

Respon

```
{
  "result": "updated",
  "_shards": {
    "total": 2
  }
}
```

Alih-alih menyertakan bidang, Anda dapat mengecualikan bidang dengan awalan `-`. `filter_path` juga mendukung wildcard:

```
POST opensearch-domain/_bulk?filter_path=-took,-items.index._*
{ "index": { "_index": "more-movies", "_id": "1" } }
{"title": "Back to the Future"}
{ "index": { "_index": "more-movies", "_id": "2" } }
{"title": "Spirited Away"}
```

Respon

```
{
  "errors": false,
  "items": [
    {
      "index": {
        "result": "updated",
        "status": 200
      }
    },
    {
      "index": {
        "result": "updated",
```

```
        "status": 200
      }
    }
  ]
}
```

Codec indeks

Codec indeks menentukan bagaimana bidang yang disimpan pada indeks dikompresi dan disimpan pada disk. Codec indeks dikendalikan oleh `index.codec` pengaturan statis, yang menentukan algoritma kompresi. Pengaturan ini berdampak pada ukuran pecahan indeks dan kinerja operasi.

Untuk daftar codec yang didukung dan karakteristik kinerjanya, lihat Codec yang didukung dalam dokumentasi. OpenSearch

Saat Anda memilih codec indeks, pertimbangkan hal berikut:

- Untuk menghindari tantangan mengubah pengaturan codec dari indeks yang ada, uji beban kerja representatif di lingkungan non-produksi sebelum menggunakan pengaturan codec baru. Untuk informasi selengkapnya, lihat Mengubah codec indeks.
- Anda tidak dapat menggunakan codec `zstd_no_dict` kompresi `zstd` dan untuk indeks K-nN atau Security Analytics.
- Migrasi ke UltraWarm instance dinonaktifkan untuk indeks zStandard.

Memuat data streaming ke OpenSearch Layanan Amazon

Anda dapat menggunakan OpenSearch Ingestion untuk langsung memuat data streaming ke domain OpenSearch Layanan Amazon Anda, tanpa perlu menggunakan solusi pihak ketiga. Untuk mengirim data ke OpenSearch Ingestion, Anda mengonfigurasi produsen data dan layanan secara otomatis mengirimkan data ke domain atau koleksi yang Anda tentukan. Untuk memulai dengan OpenSearch Ingestion, lihat. the section called “Tutorial: Menyerap data ke dalam koleksi”

Anda masih dapat menggunakan sumber lain untuk memuat data streaming, seperti Amazon Data Firehose dan Amazon CloudWatch Logs, yang memiliki dukungan bawaan untuk OpenSearch Layanan. Lainnya, seperti Amazon S3, Amazon Kinesis Data Streams, dan Amazon DynamoDB, gunakan fungsi AWS Lambda sebagai event handler. Fungsi Lambda menanggapi data baru dengan memproses dan streaming data itu ke domain Anda.

> **Note**
>
> Lambda mendukung beberapa bahasa pemrograman populer dan tersedia di sebagian besar. Wilayah AWSUntuk informasi selengkapnya, lihat Memulai Lambda di PanduanAWS Lambda Pengembang dan titik akhirAWS layanan di. Referensi Umum AWS

Topik

Memuat data streaming dari OpenSearch Ingestion

Anda dapat menggunakan Amazon OpenSearch Ingestion untuk memuat data ke domain OpenSearch Layanan. Anda mengonfigurasi produsen data Anda untuk mengirim data ke OpenSearch Ingestion, dan secara otomatis mengirimkan data ke koleksi yang Anda tentukan. Anda juga dapat mengonfigurasi OpenSearch Ingestion untuk mengubah data Anda sebelum mengirimkannya. Untuk informasi selengkapnya, lihat OpenSearch Tertelan Amazon.

Memuat data streaming dari Amazon S3

Anda dapat menggunakan Lambda untuk mengirim data ke domain OpenSearch Layanan Anda dari Amazon S3. Data baru yang tiba di bucket S3 memicu notifikasi peristiwa untuk Lambda, yang kemudian menjalankan kode kustom Anda untuk melakukan pengindeksan.

Metode data streaming ini sangat fleksibel. Anda dapat mengindeks metadata objek, atau jika objek plaintext, mengurai dan mengindeks beberapa elemen dari tubuh objek. Bagian ini mencakup beberapa kode sampel Python yang kurang modern yang menggunakan ekspresi reguler untuk mengurai file log dan mengindeks kecocokan.

Prasyarat

Sebelum melanjutkan, Anda harus memiliki sumber daya berikut.

Prasyarat	Deskripsi
Bucket Amazon S3	Untuk informasi selengkapnya, lihat Membuat bucket S3 pertama Anda di Panduan Pengguna Layanan Penyimpanan Sederhana Amazon. Bucket harus berada di Wilayah yang sama dengan domain OpenSearch Layanan Anda.
OpenSearch Domain layanan	Tujuan untuk data setelah fungsi Lambda Anda memprosesnya. Untuk informasi selengkapnya, lihat the section called “ Membuat domain OpenSearch Layanan”.

Membuat paket deployment Lambda

Paket deployment adalah file ZIP atau JAR yang berisi kode Anda dan dependensinya. Bagian ini mencakup kode sampel Python. Untuk bahasa pemrograman lainnya, lihat Paket penyebaran Lambda di Panduan Pengembang.AWS Lambda

1. Buatlah sebuah direktori. Dalam sampel ini, kita menggunakan nama `s3-to-opensearch`.
2. Buat file dalam direktori bernama`sample.py`:

```
import boto3
import re
import requests
from requests_aws4auth import AWS4Auth

region = '' # e.g. us-west-1
service = 'es'
credentials = boto3.Session().get_credentials()
awsauth = AWS4Auth(credentials.access_key, credentials.secret_key, region, service,
 session_token=credentials.token)

host = '' # the OpenSearch Service domain, e.g. https://search-mydomain.us-
west-1.es.amazonaws.com
index = 'lambda-s3-index'
datatype = '_doc'
```

```
url = host + '/' + index + '/' + datatype

headers = { "Content-Type": "application/json" }

s3 = boto3.client('s3')

# Regular expressions used to parse some simple log lines
ip_pattern = re.compile('(\d+\.\d+\.\d+\.\d+)')
time_pattern = re.compile('\[(\d+\/\w\w\w\/\d\d\d\d:\d\d:\d\d:\d\d\s-\d\d\d\d)\]')
message_pattern = re.compile('\"(.+)\"')

# Lambda execution starts here
def handler(event, context):
    for record in event['Records']:

        # Get the bucket name and key for the new file
        bucket = record['s3']['bucket']['name']
        key = record['s3']['object']['key']

        # Get, read, and split the file into lines
        obj = s3.get_object(Bucket=bucket, Key=key)
        body = obj['Body'].read()
        lines = body.splitlines()

        # Match the regular expressions to each line and index the JSON
        for line in lines:
            line = line.decode("utf-8")
            ip = ip_pattern.search(line).group(1)
            timestamp = time_pattern.search(line).group(1)
            message = message_pattern.search(line).group(1)

            document = { "ip": ip, "timestamp": timestamp, "message": message }
            r = requests.post(url, auth=awsauth, json=document, headers=headers)
```

Edit variabel untuk `region` dan `host`.

3. Instal pip jika Anda belum melakukannya, lalu instal dependensi ke direktori baru: `package`

```
cd s3-to-opensearch

pip install --target ./package requests
pip install --target ./package requests_aws4auth
```

Semua lingkungan eksekusi Lambda telah menginstal Boto3, sehingga Anda tidak perlu memasukkannya ke dalam paket deployment.

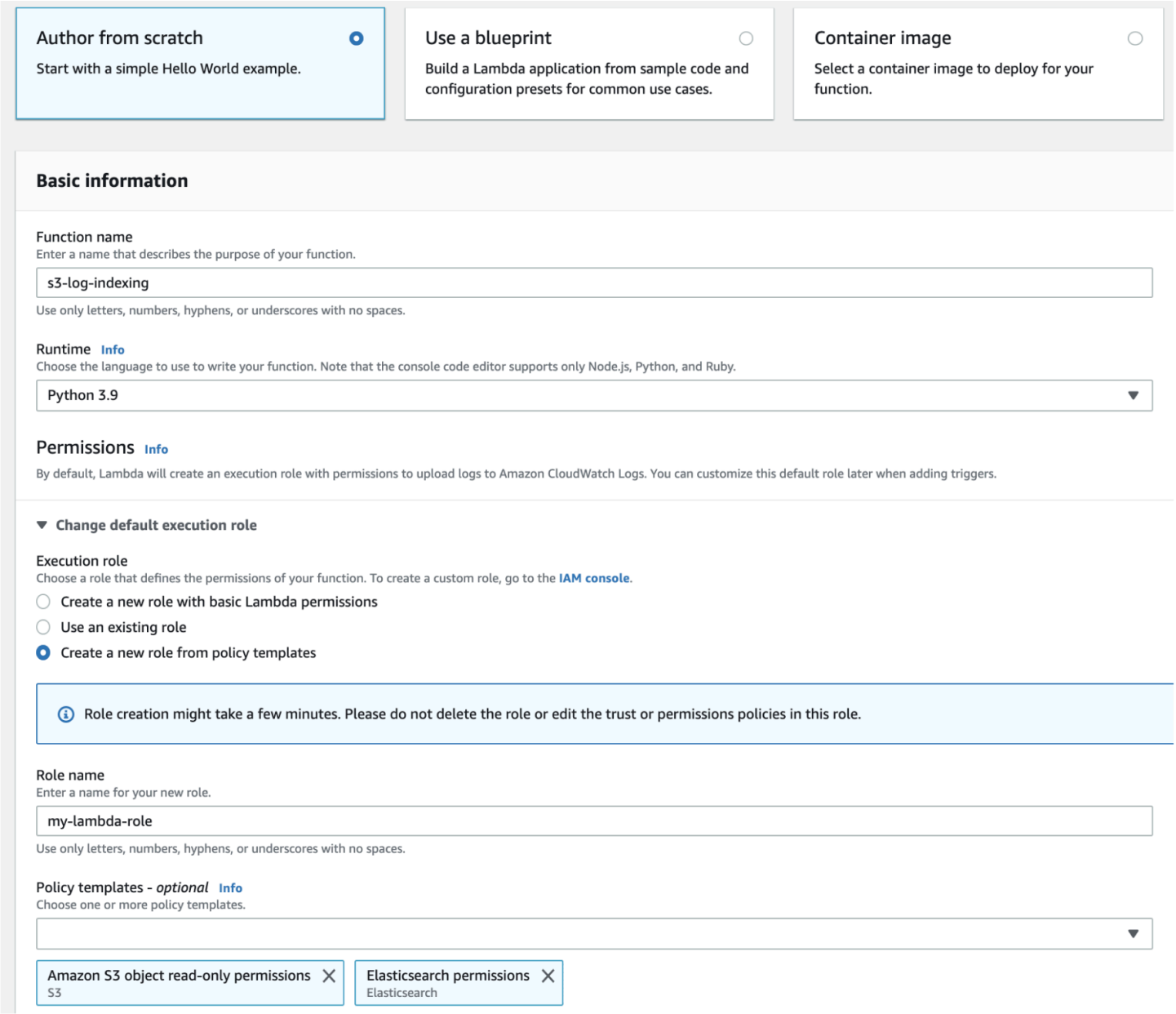
4. Paket kode aplikasi dan dependensi:

```
cd package
zip -r ../lambda.zip .

cd ..
zip -g lambda.zip sample.py
```

Buat fungsi Lambda

Setelah Anda membuat paket deployment, Anda dapat membuat fungsi Lambda. Saat Anda membuat fungsi, pilih nama, runtime (misalnya, Python 3.8), dan peran IAM. IAM role mendefinisikan izin untuk fungsi Anda. Untuk petunjuk mendetail, lihat Membuat fungsi Lambda dengan konsol di PanduanAWS Lambda Pengembang.

Contoh ini mengasumsikan Anda menggunakan konsol. Pilih Python 3.9 dan peran yang memiliki izin baca S3 dan izin menulis OpenSearch Layanan, seperti yang ditunjukkan pada gambar berikut:



Setelah Anda membuat fungsi, Anda harus menambahkan pemicu. Untuk contoh ini, kita ingin kode berjalan setiap kali file log tiba di bucket S3:

1. Pilih Tambah pemicu dan pilih S3.
2. Pilih bucket Anda.
3. Untuk Jenis peristiwa, pilih TEMPATKAN.
4. Untuk Prefiks, ketik `logs/`.
5. Untuk Sufiks, ketik `.log`.
6. Akui peringatan pemanggilan rekursif dan pilih Tambah.

Akhirnya, Anda dapat mengunggah paket deployment Anda:

1. Pilih Unggah dari dan file.zip, lalu ikuti petunjuk untuk mengunggah paket penerapan Anda.
2. Setelah unggahan selesai, edit pengaturan Runtime dan ubah Handler menjadi. `sample.handler` Pengaturan ini memberitahu Lambda file (`sample.py`) dan metode (`handler`) yang harus dijalankan setelah pemicu aktif.

Pada titik ini, Anda memiliki satu set lengkap sumber daya: bucket untuk file log, fungsi yang berjalan setiap kali file log ditambahkan ke bucket, kode yang melakukan penguraian dan pengindeksan, dan domain OpenSearch Layanan untuk pencarian dan visualisasi.

Menguji fungsi Lambda

Setelah Anda membuat fungsi, Anda dapat mengujinya dengan mengunggah file ke bucket Amazon S3. Buat file bernama `sample.log` dengan menggunakan baris log sampel berikut:

```
12.345.678.90 - [10/Oct/2000:13:55:36 -0700] "PUT /some-file.jpg"
12.345.678.91 - [10/Oct/2000:14:56:14 -0700] "GET /some-file.jpg"
```

Unggah file ke folder `logs` dari bucket S3 Anda. Untuk petunjuk, lihat Mengunggah objek ke bucket Anda di Panduan Pengguna Layanan Penyimpanan Sederhana Amazon.

Kemudian gunakan konsol OpenSearch Layanan atau OpenSearch Dasbor untuk memverifikasi bahwa `lambda-s3-index` indeks berisi dua dokumen. Anda juga dapat membuat permintaan pencarian standar:

```
GET https://domain-name/lambda-s3-index/_search?pretty
{
  "hits" : {
    "total" : 2,
    "max_score" : 1.0,
    "hits" : [
      {
        "_index" : "lambda-s3-index",
        "_type" : "_doc",
        "_id" : "vTYXaWIBJWV_TTkEuSDg",
        "_score" : 1.0,
        "_source" : {
          "ip" : "12.345.678.91",
          "message" : "GET /some-file.jpg",
```

```
          "timestamp" : "10/Oct/2000:14:56:14 -0700"
        }
      },
      {
        "_index" : "lambda-s3-index",
        "_type" : "_doc",
        "_id" : "vjYmaWIBJWV_TTkEuCAB",
        "_score" : 1.0,
        "_source" : {
          "ip" : "12.345.678.90",
          "message" : "PUT /some-file.jpg",
          "timestamp" : "10/Oct/2000:13:55:36 -0700"
        }
      }
    ]
  }
}
```

Memuat data streaming dari Amazon Kinesis Data Streams

Anda dapat memuat data streaming dari Kinesis Data OpenSearch Streams ke Layanan. Data baru yang tiba di aliran data memicu notifikasi peristiwa untuk Lambda, yang kemudian menjalankan kode kustom Anda untuk melakukan pengindeksan. Bagian ini mencakup kode sampel Python yang kurang modern.

Prasyarat

Sebelum melanjutkan, Anda harus memiliki sumber daya berikut.

Prasyarat	Deskripsi
Amazon Kinesis Data Stream	Sumber peristiwa untuk fungsi Lambda Anda. Untuk mempelajari selengkapnya, lihat Kinesis Data Streams.
OpenSearch Layanan Domain	Tujuan untuk data setelah fungsi Lambda Anda memprosesnya. Lihat informasi yang lebih lengkap di the section called “ Membuat domain OpenSearch Layanan”
IAM Role	Peran ini harus memiliki izin dasar OpenSearch Layanan, Kinesis, dan Lambda, seperti berikut ini:

Prasyarat	Deskripsi

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Effect": "Allow",
      "Action": [
        "es:ESHttpPost",
        "es:ESHttpPut",
        "logs:CreateLogGroup",
        "logs:CreateLogStream",
        "logs:PutLogEvents",
        "kinesis:GetShardIterator",
        "kinesis:GetRecords",
        "kinesis:DescribeStream",
        "kinesis:ListStreams"
      ],
      "Resource": "*"
    }
  ]
}
```

Peran harus memiliki hubungan kepercayaan berikut:

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

Untuk mempelajari selengkapnya, lihat Membuat peran IAM di Panduan Pengguna IAM.

Buat fungsi Lambda

Ikuti instruksi di the section called “Membuat paket deployment Lambda”, tapi buat sebuah direktori bernama `kinesis-to-opensearch` dan gunakan kode berikut untuk `sample.py`:

```
import base64
import boto3
import json
import requests
from requests_aws4auth import AWS4Auth

region = '' # e.g. us-west-1
service = 'es'
credentials = boto3.Session().get_credentials()
awsauth = AWS4Auth(credentials.access_key, credentials.secret_key, region, service,
 session_token=credentials.token)

host = '' # the OpenSearch Service domain, e.g. https://search-mydomain.us-
west-1.es.amazonaws.com
index = 'lambda-kine-index'
datatype = '_doc'
url = host + '/' + index + '/' + datatype + '/'

headers = { "Content-Type": "application/json" }

def handler(event, context):
    count = 0
    for record in event['Records']:
        id = record['eventID']
        timestamp = record['kinesis']['approximateArrivalTimestamp']

        # Kinesis data is base64-encoded, so decode here
        message = base64.b64decode(record['kinesis']['data'])

        # Create the JSON document
        document = { "id": id, "timestamp": timestamp, "message": message }
        # Index the document
        r = requests.put(url + id, auth=awsauth, json=document, headers=headers)
        count += 1
    return 'Processed ' + str(count) + ' items.'
```

Edit variabel untuk `region` dan `host`.

Instal pip jika Anda belum melakukannya, maka gunakan perintah berikut untuk menginstal dependensi Anda:

```
cd kinesis-to-opensearch

pip install --target ./package requests
pip install --target ./package requests_aws4auth
```

Kemudian ikuti instruksi di the section called “Buat fungsi Lambda”, namun tentukan IAM role dari the section called “Prasyarat” dan pengaturan berikut untuk pemicu:

- Pengaliran Kinesis: Pengaliran Kinesis Anda
- Ukuran Batch: 100
- Posisi awal: Potong cakrawala

Untuk mempelajari lebih lanjut, lihat Apa itu Amazon Kinesis Data Streams? di Panduan Pengembang Amazon Kinesis Data Streams.

Pada titik ini, Anda memiliki satu set lengkap sumber daya: aliran data Kinesis, fungsi yang berjalan setelah aliran menerima data baru dan mengindeks data tersebut, dan domain OpenSearch Layanan untuk pencarian dan visualisasi.

Uji Fungsi Lambda

Setelah Anda membuat fungsi, Anda dapat mengujinya dengan menambahkan catatan baru ke aliran data dengan menggunakan AWS CLI:

```
aws kinesis put-record --stream-name test --data "My test data." --partition-key
 partitionKey1 --region us-west-1
```

Kemudian gunakan konsol OpenSearch Layanan atau OpenSearch Dasbor untuk memverifikasi bahwa `lambda-kine-index` berisi dokumen. Anda juga dapat menggunakan permintaan berikut:

```
GET https://domain-name/lambda-kine-index/_search
{
  "hits" : [
    {
      "_index": "lambda-kine-index",
```

```
      "_type": "_doc",
      "_id":
 "shardId-000000000000:495835116157626994950129608214214566865294366804960870 42",
      "_score": 1,
      "_source": {
        "timestamp": 1523648740.051,
        "message": "My test data.",
        "id":
 "shardId-000000000000:495835116157626994950129608214214566865294366804960870 42"
      }
    }
  ]
}
```

Memuat data streaming dari Amazon DynamoDB

Anda dapat menggunakan AWS Lambda untuk mengirim data ke domain OpenSearch Layanan Anda dari Amazon DynamoDB. Data baru yang tiba di basis data memicu notifikasi peristiwa untuk Lambda, yang kemudian menjalankan kode kustom Anda untuk melakukan pengindeksan.

Prasyarat

Sebelum melanjutkan, Anda harus memiliki sumber daya berikut.

Prasyarat	Deskripsi
Tabel DynamoDB	Tabel berisi data sumber Anda. Untuk informasi selengkapnya, lihat Operasi Dasar pada Tabel DynamoDB di Panduan Pengembang Amazon DynamoDB. Tabel harus berada di Wilayah yang sama dengan domain OpenSearch Layanan Anda dan memiliki aliran yang disetel ke Gambar baru. Untuk mempelajari selengkapnya, lihat Mengaktifkan Pengaliran.
OpenSearch Domain layanan	Tujuan untuk data setelah fungsi Lambda Anda memprosesnya. Untuk informasi selengkapnya, lihat the section called “ Membuat domain OpenSearch Layanan”.
IAM Role	Peran ini harus memiliki izin eksekusi OpenSearch Service, DynamoDB, dan Lambda dasar, seperti berikut ini:

Prasyarat	Deskripsi

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Effect": "Allow",
      "Action": [
        "es:ESHttpPost",
        "es:ESHttpPut",
        "dynamodb:DescribeStream",
        "dynamodb:GetRecords",
        "dynamodb:GetShardIterator",
        "dynamodb:ListStreams",
        "logs:CreateLogGroup",
        "logs:CreateLogStream",
        "logs:PutLogEvents"
      ],
      "Resource": "*"
    }
  ]
}
```

Peran harus memiliki hubungan kepercayaan berikut:

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

Untuk mempelajari selengkapnya, lihat Membuat peran IAM di Panduan Pengguna IAM.

Buat fungsi Lambda

Ikuti instruksi di the section called “Membuat paket deployment Lambda”, tapi buat sebuah direktori bernama `ddb-to-opensearch` dan gunakan kode berikut untuk `sample.py`:

```
import boto3
import requests
from requests_aws4auth import AWS4Auth

region = '' # e.g. us-east-1
service = 'es'
credentials = boto3.Session().get_credentials()
awsauth = AWS4Auth(credentials.access_key, credentials.secret_key, region, service,
 session_token=credentials.token)

host = '' # the OpenSearch Service domain, e.g. https://search-mydomain.us-
west-1.es.amazonaws.com
index = 'lambda-index'
datatype = '_doc'
url = host + '/' + index + '/' + datatype + '/'

headers = { "Content-Type": "application/json" }

def handler(event, context):
    count = 0
    for record in event['Records']:
        # Get the primary key for use as the OpenSearch ID
        id = record['dynamodb']['Keys']['id']['S']

        if record['eventName'] == 'REMOVE':
            r = requests.delete(url + id, auth=awsauth)
        else:
            document = record['dynamodb']['NewImage']
            r = requests.put(url + id, auth=awsauth, json=document, headers=headers)
        count += 1
    return str(count) + ' records processed.'
```

Edit variabel untuk `region` dan `host`.

Instal pip jika Anda belum melakukannya, maka gunakan perintah berikut untuk menginstal dependensi Anda:

```
cd ddb-to-opensearch
```

```
pip install --target ./package requests
pip install --target ./package requests_aws4auth
```

Kemudian ikuti instruksi di the section called “Buat fungsi Lambda”, namun tentukan IAM role dari the section called “Prasyarat” dan pengaturan berikut untuk pemicu:

- Tabel: Tabel DynamoDB Anda
- Ukuran Batch: 100
- Posisi awal: Potong cakrawala

Untuk mempelajari lebih lanjut, lihat Memproses Item Baru dengan DynamoDB Streams dan Lambda di Panduan Pengembang Amazon DynamoDB.

Pada titik ini, Anda memiliki satu set lengkap sumber daya: tabel DynamoDB untuk data sumber Anda, aliran perubahan DynamoDB pada tabel, fungsi yang berjalan setelah data sumber Anda berubah dan mengindeks perubahan tersebut, dan domain Layanan untuk pencarian dan visualisasi. OpenSearch

Tes fungsi Lambda

Setelah Anda membuat fungsi, Anda dapat mengujinya dengan menambahkan item baru ke tabel DynamoDB dengan menggunakan AWS CLI:

```
aws dynamodb put-item --table-name test --item '{"director": {"S": "Kevin
 Costner"},"id": {"S": "00001"},"title": {"S": "The Postman"}}' --region us-west-1
```

Kemudian gunakan konsol OpenSearch Layanan atau OpenSearch Dasbor untuk memverifikasi bahwa `lambda-index` berisi dokumen. Anda juga dapat menggunakan permintaan berikut:

```
GET https://domain-name/lambda-index/_doc/00001
{
    "_index": "lambda-index",
    "_type": "_doc",
    "_id": "00001",
    "_version": 1,
    "found": true,
    "_source": {
        "director": {
```

```
            "S": "Kevin Costner"
        },
        "id": {
            "S": "00001"
        },
        "title": {
            "S": "The Postman"
        }
    }
}
```

Memuat data streaming dari Amazon Data Firehose

Firehose mendukung OpenSearch Layanan sebagai tujuan pengiriman. Untuk petunjuk tentang cara memuat data streaming ke OpenSearch Layanan, lihat Membuat Aliran Pengiriman Firehose Data Kinesis OpenSearch dan Memilih Layanan untuk Tujuan Anda di Panduan Pengembang Amazon Data Firehose.

Sebelum memuat data ke OpenSearch Layanan, Anda mungkin perlu melakukan transformasi pada data. Untuk mempelajari selengkapnya tentang penggunaan fungsi Lambda untuk menjalankan tugas ini, lihat Transformasi Data Amazon Kinesis Data Firehose dalam panduan yang sama.

Saat Anda mengonfigurasi aliran pengiriman, Firehose menampilkan peran IAM “sekali klik” yang memberikan akses sumber daya yang diperlukan untuk mengirim data ke OpenSearch Layanan, mencadangkan data di Amazon S3, dan mengubah data menggunakan Lambda. Karena kompleksitas yang terlibat dalam menciptakan peran semacam itu secara manual, sebaiknya gunakan peran yang disediakan.

Memuat data streaming dari Amazon CloudWatch

Anda dapat memuat data streaming dari CloudWatch Log ke domain OpenSearch Layanan Anda dengan menggunakan langganan CloudWatch Log. Untuk informasi tentang CloudWatch langganan Amazon, lihat Pemrosesan data log secara real-time dengan langganan. Untuk informasi konfigurasi, lihat data Streaming CloudWatch Log ke OpenSearch Layanan Amazon di Panduan CloudWatch Pengembang Amazon.

Memuat data streaming dari AWS IoT

Anda dapat mengirim data dari AWS IoT menggunakan aturan. Untuk mempelajari lebih lanjut, lihat OpenSearchtindakan di PanduanAWS IoT Pengembang.

Memuat data ke Amazon OpenSearch Service dengan Logstash

Versi sumber terbuka dari Logstash (Logstash OSS) menyediakan cara mudah untuk menggunakan API massal untuk mengunggah data ke domain Amazon OpenSearch Service Anda. Layanan ini mendukung semua plugin input Logstash standar, termasuk plugin input Amazon S3. OpenSearch Layanan mendukung plugin logstash-output-opensearchoutput, yang mendukung otentikasi dasar dan kredensi IAM. Plugin ini bekerja dengan versi 8.1 dan lebih rendah dari Logstash OSS.

Konfigurasi

Konfigurasi Logstash bervariasi berdasarkan jenis autentikasi domain Anda.

Apa pun metode otentikasi yang Anda gunakan, Anda harus menyetel`ecs_compatibility` ke`disabled` bagian output dari file konfigurasi. Logstash 8.0 memperkenalkan perubahan melanggar di mana semua plugin dijalankan dalam modus kompatibilitas ECS secara default. Anda harus mengganti nilai default untuk mempertahankan perilaku lama.

Konfigurasi kontrol akses detail

Jika domain OpenSearch Service Anda menggunakan kontrol akses detail dengan autentikasi dasar HTTP, konfigurasi mirip dengan OpenSearch klaster lainnya. File konfigurasi contoh ini mengambil input dari versi sumber terbuka dari Filebeat (Filebeat OSS):

```
input {
  beats  {
    port => 5044
  }
}

output {
  opensearch {
    hosts       => "https://domain-endpoint:443"
    user        => "my-username"
    password    => "my-password"
    index       => "logstash-logs-%{+YYYY.MM.dd}"
    ecs_compatibility => disabled
    ssl_certificate_verification => false
  }
}
```

Konfigurasi bervariasi berdasarkan aplikasi Beats dan kasus penggunaan, tetapi konfigurasi Filebeat OSS Anda mungkin terlihat seperti ini:

```
filebeat.inputs:
- type: log
  enabled: true
  paths:
    - /path/to/logs/dir/*.log
filebeat.config.modules:
  path: ${path.config}/modules.d/*.yml
  reload.enabled: false
setup.ilm.enabled: false
setup.ilm.check_exists: false
setup.template.settings:
  index.number_of_shards: 1
output.logstash:
  hosts: ["logstash-host:5044"]
```

Konfigurasi IAM

Jika domain Anda menggunakan kebijakan akses domain berbasis IAM atau kontrol akses detail dengan pengguna utama, Anda harus menandatangani semua permintaan ke OpenSearch Layanan menggunakan kredensi IAM. Kebijakan berbasis identitas berikut memberikan semua permintaan HTTP ke subsumber daya domain Anda.

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Effect": "Allow",
      "Action": [
        "es:ESHttp*"
      ],
      "Resource": "arn:aws:es:region:aws-account-id:domain/domain-name/*"
    }
  ]
}
```

Untuk mengatur konfigurasi Logstash Anda, ubah file konfigurasi Anda agar dapat menggunakan plugin untuk outputnya. File konfigurasi contoh ini mengambil input dari file dalam bucket S3:

```
input {
```

```
  s3 {
    bucket => "my-s3-bucket"
    region => "us-east-1"
  }
}

output {
  opensearch {
    hosts => ["domain-endpoint:443"]
    auth_type => {
      type => 'aws_iam'
      aws_access_key_id => 'your-access-key'
      aws_secret_access_key => 'your-secret-key'
      region => 'us-east-1'
      }
      index  => "logstash-logs-%{+YYYY.MM.dd}"
      ecs_compatibility => disabled
  }
}
```

Jika Anda tidak ingin memberikan kredensi IAM Anda dalam file konfigurasi, Anda dapat mengekspornya (atau menjalankan`aws configure`):

```
export AWS_ACCESS_KEY_ID="your-access-key"
export AWS_SECRET_ACCESS_KEY="your-secret-key"
export AWS_SESSION_TOKEN="your-session-token"
```

Jika domain OpenSearch Layanan Anda adalah dalam VPC, mesin Logstash OSS harus dapat terhubung ke VPC dan memiliki akses ke domain melalui grup keamanan VPC. Untuk informasi selengkapnya, lihat the section called “Tentang kebijakan akses pada domain VPC”.

Mencari data di Amazon OpenSearch Service

Ada beberapa metode umum untuk mencari dokumen di Amazon OpenSearch Service, termasuk pencarian URI dan pencarian isi permintaan. OpenSearch Layanan menawarkan fungsionalitas tambahan yang meningkatkan pengalaman pencarian, seperti paket khusus, dukungan SQL, dan pencarian asinkron. Untuk referensi API OpenSearch penelusuran yang komprehensif, lihat OpenSearch dokumentasinya.

> **Note**
>
> Contoh permintaan berikut bekerja dengan OpenSearch API. Beberapa permintaan mungkin tidak berfungsi dengan versi Elasticsearch yang lebih lama.

Topik

Pencarian URI

Pencarian Universal Resource Identifier (URI) adalah bentuk pencarian yang paling sederhana. Dalam pencarian URI, Anda menentukan kueri sebagai parameter permintaan HTTP:

```
GET https://search-my-domain.us-west-1.es.amazonaws.com/_search?q=house
```

Respons sampel mungkin terlihat seperti berikut ini:


```
{
  "took": 25,
  "timed_out": false,
  "_shards": {
    "total": 10,
    "successful": 10,
    "skipped": 0,
    "failed": 0
  },
  "hits": {
    "total": {
      "value": 85,
      "relation": "eq",
    },
    "max_score": 6.6137657,
    "hits": [
      {
        "_index": "movies",
        "_type": "movie",
        "_id": "tt0077975",
        "_score": 6.6137657,
        "_source": {
          "directors": [
            "John Landis"
          ],
          "release_date": "1978-07-27T00:00:00Z",
          "rating": 7.5,
          "genres": [
            "Comedy",
            "Romance"
          ],
          "image_url": "http://ia.media-imdb.com/images/M/
MV5BMTY2OTQxNTc1OF5BMl5BanBnXkFtZTYwNjA3NjI5._V1_SX400_.jpg",
          "plot": "At a 1962 College, Dean Vernon Wormer is determined to expel the
 entire Delta Tau Chi Fraternity, but those troublemakers have other plans for him.",
          "title": "Animal House",
          "rank": 527,
          "running_time_secs": 6540,
          "actors": [
```

```
            "John Belushi",
            "Karen Allen",
            "Tom Hulce"
          ],
          "year": 1978,
          "id": "tt0077975"
        }
      },
      ...
    ]
  }
}
```

Secara default, kueri ini mencari semua bidang dari semua indeks untuk istilah rumah. Untuk mempersempit pencarian, tentukan indeks (`movies`) dan bidang dokumen (`title`) di URI:

```
GET https://search-my-domain.us-west-1.es.amazonaws.com/movies/_search?q=title:house
```

Anda dapat memasukkan parameter tambahan dalam permintaan, tetapi parameter yang didukung hanya menyediakan sebagian kecil dari opsi OpenSearch pencarian. Permintaan berikut mengembalikan 20 hasil (bukan defaultnya yang 10) dan mengurutkan menurut tahun (dibandingkan dengan `_score`):

```
GET https://search-my-domain.us-west-1.es.amazonaws.com/movies/_search?
q=title:house&size=20&sort=year:desc
```

Pencarian isi permintaan

Untuk melakukan pencarian yang lebih kompleks, gunakan isi permintaan HTTP dan bahasa OpenSearch khusus domain (DSL) untuk kueri. Kueri DSL memungkinkan Anda menentukan berbagai opsi OpenSearch pencarian.

> **Note**
>
> Anda tidak dapat menyertakan karakter khusus Unicode dalam nilai bidang teks, atau nilainya akan diuraikan sebagai beberapa nilai yang dipisahkan oleh karakter khusus. Penguraian yang salah ini dapat menyebabkan penyaringan dokumen yang tidak disengaja dan berpotensi membahayakan kontrol atas aksesnya. Untuk informasi selengkapnya, lihat Catatan tentang karakter khusus Unicode di bidang teks dalam OpenSearch dokumentasi.

Kueri `match` berikut ini serupa dengan contoh pencarian URI final:

```
POST https://search-my-domain.us-west-1.es.amazonaws.com/movies/_search
{
  "size": 20,
  "sort": {
    "year": {
      "order": "desc"
    }
  },
  "query": {
    "query_string": {
      "default_field": "title",
      "query": "house"
    }
  }
}
```

> **Note**
>
> API `_search` menerima HTTP `GET` dan `POST` untuk pencarian isi permintaan, tetapi tidak semua klien HTTP mendukung menambahkan isi permintaan ke permintaan `GET`.`POST` adalah pilihan yang lebih universal.

Dalam banyak kasus, Anda mungkin ingin mencari beberapa bidang, tetapi tidak semua bidang. Gunakan kueri `multi_match`:

```
POST https://search-my-domain.us-west-1.es.amazonaws.com/movies/_search
{
  "size": 20,
  "query": {
    "multi_match": {
      "query": "house",
      "fields": ["title", "plot", "actors", "directors"]
    }
  }
}
```

Bidang pendorong

Anda dapat meningkatkan relevansi pencarian dengan “meningkatkan” bidang tertentu. Boost adalah pengganda yang menimbang kecocokan dalam satu bidang lebih berat daripada kecocokan di bidang lain. Pada contoh berikut, kecocokan untuk john di bidang pengaruh `title` `_score` dua kali lebih banyak dari kecocokan di bidang `plot` dan empat kali lebih banyak kecocokan di bidang `actors` atau `directors`. Hasilnya adalah bahwa film seperti John Wick dan John Carter berada di dekat puncak hasil pencarian, dan film yang dibintangi John Travolta berada di dekat bagian bawah.

```
POST https://search-my-domain.us-west-1.es.amazonaws.com/movies/_search
{
  "size": 20,
  "query": {
    "multi_match": {
      "query": "john",
      "fields": ["title^4", "plot^2", "actors", "directors"]
    }
  }
}
```

Penyorotan hasil pencarian

`highlight`Opsi memberitahu OpenSearch untuk mengembalikan objek tambahan di dalam `hits` array jika query cocok dengan satu atau beberapa bidang:

```
POST https://search-my-domain.us-west-1.es.amazonaws.com/movies/_search
{
  "size": 20,
  "query": {
    "multi_match": {
      "query": "house",
      "fields": ["title^4", "plot^2", "actors", "directors"]
    }
  },
  "highlight": {
    "fields": {
      "plot": {}
    }
  }
}
```

Jika kueri cocok dengan isi bidang `plot`, temuan mungkin terlihat seperti berikut ini:


```
{
  "_index": "movies",
  "_type": "movie",
  "_id": "tt0091541",
  "_score": 11.276199,
  "_source": {
    "directors": [
      "Richard Benjamin"
    ],
    "release_date": "1986-03-26T00:00:00Z",
    "rating": 6,
    "genres": [
      "Comedy",
      "Music"
    ],
    "image_url": "http://ia.media-imdb.com/images/M/
MV5BMTIzODEzODE2OF5BMl5BanBnXkFtZTcwNjQ3ODcyMQ@@._V1_SX400_.jpg",
    "plot": "A young couple struggles to repair a hopelessly dilapidated house.",
    "title": "The Money Pit",
    "rank": 4095,
    "running_time_secs": 5460,
    "actors": [
      "Tom Hanks",
      "Shelley Long",
      "Alexander Godunov"
    ],
    "year": 1986,
    "id": "tt0091541"
  },
  "highlight": {
    "plot": [
      "A young couple struggles to repair a hopelessly dilapidated <em>house</em>."
    ]
  }
}
```


Secara default, OpenSearch membungkus string yang cocok dalam tag, menyediakan hingga 100 karakter konteks di sekitar pertandingan, dan memecah konten menjadi kalimat dengan mengidentifikasi tanda baca, spasi, tab, dan jeda baris. Semua pengaturan ini dapat disesuaikan:

```
POST https://search-my-domain.us-west-1.es.amazonaws.com/movies/_search
```

```
{
  "size": 20,
  "query": {
    "multi_match": {
      "query": "house",
      "fields": ["title^4", "plot^2", "actors", "directors"]
    }
  },
  "highlight": {
    "fields": {
      "plot": {}
    },
    "pre_tags": "<strong>",
    "post_tags": "</strong>",
    "fragment_size": 200,
    "boundary_chars": ".,!? "
  }
}
```

Jumlah API

Jika Anda tidak tertarik dengan isi dokumen Anda dan hanya ingin mengetahui jumlah kecocokan, Anda dapat menggunakan API `_count` bukan API `_search`. Permintaan berikut menggunakan kueri `query_string` untuk mengidentifikasi komedi romantis:

```
POST https://search-my-domain.us-west-1.es.amazonaws.com/movies/_count
{
  "query": {
    "query_string": {
      "default_field": "genres",
      "query": "romance AND comedy"
    }
  }
}
```

Respons sampel mungkin terlihat seperti berikut ini:

```
{
  "count": 564,
  "_shards": {
    "total": 5,
    "successful": 5,
```

```
    "skipped": 0,
    "failed": 0
  }
}
```

Pemberian nomor halaman hasil pencarian

Jika Anda perlu menampilkan sejumlah besar hasil pencarian, Anda dapat menerapkan pagination menggunakan beberapa metode berbeda.

Titik waktu

Fitur point in time (PIT) adalah jenis pencarian yang memungkinkan Anda menjalankan kueri berbeda terhadap kumpulan data yang diperbaiki tepat waktu. Ini adalah metode pagination yang disukai OpenSearch, terutama untuk penomoran halaman yang dalam. Anda dapat menggunakan PIT dengan OpenSearch Layanan versi 2.5 dan yang lebih baru. Untuk informasi lebih lanjut tentang PIT, lihat???.

`from`Dan **`size`** parameter

Cara termudah untuk membuat halaman adalah dengan `size` parameter `from` dan. Permintaan berikut mengembalikan hasil 20—39 dari daftar hasil pencarian yang diindeks nol:

```
POST https://search-my-domain.us-west-1.es.amazonaws.com/movies/_search
{
  "from": 20,
  "size": 20,
  "query": {
    "multi_match": {
      "query": "house",
      "fields": ["title^4", "plot^2", "actors", "directors"]
    }
  }
}
```

Untuk informasi selengkapnya tentang pagination penelusuran, lihat hasil Paginate di dokumentasi. OpenSearch

Bahasa Kueri Dasbor

Anda dapat menggunakan Dashboards Query Language (DQL) untuk mencari data dan visualisasi di Dasbor. OpenSearch DQL menggunakan empat jenis kueri utama: istilah, Boolean, tanggal dan rentang, dan bidang bersarang.

Permintaan persyaratan

Kueri istilah mengharuskan Anda menentukan istilah yang Anda cari.

Untuk melakukan kueri istilah, masukkan yang berikut ini:

```
host:www.example.com
```

Kueri Boolean

Anda dapat menggunakan operator Boolean`AND`,`or`, dan `not` untuk menggabungkan beberapa query.

Untuk melakukan kueri Boolean, tempel berikut ini:

```
host.keyword:www.example.com and response.keyword:200
```

Kueri tanggal dan rentang

Anda dapat menggunakan kueri tanggal dan rentang untuk menemukan tanggal sebelum atau sesudah kueri Anda.

- >menunjukkan pencarian untuk tanggal setelah tanggal yang Anda tentukan.
- <menunjukkan pencarian untuk tanggal sebelum tanggal yang Anda tentukan.

```
@timestamp > "2020-12-14T09:35:33"
```

Kueri bidang bersarang

Jika Anda memiliki dokumen dengan bidang bersarang, Anda harus menentukan bagian mana dari dokumen yang ingin Anda ambil. Berikut ini adalah contoh dokumen yang berisi bidang bersarang:

```
{"NBA players":[
```

```
    {"player-name": "Lebron James",
      "player-position": "Power forward",
      "points-per-game": "30.3"
    },
    {"player-name": "Kevin Durant",
      "player-position": "Power forward",
      "points-per-game": "27.1"
    },
    {"player-name": "Anthony Davis",
      "player-position": "Power forward",
      "points-per-game": "23.2"
    },
    {"player-name": "Giannis Antetokounmpo",
      "player-position": "Power forward",
      "points-per-game":"29.9"
    }
  ]
}
```

Untuk mengambil bidang tertentu menggunakan DQL, tempel yang berikut ini:

```
NBA players: {player-name: Lebron James}
```

Untuk mengambil beberapa objek dari dokumen bersarang, tempel yang berikut ini:

```
NBA players: {player-name: Lebron James} and NBA players: {player-name: Giannis
 Antetokounmpo}
```

Untuk mencari dalam rentang, tempel yang berikut ini:

```
NBA players: {player-name: Lebron James} and NBA players: {player-name: Giannis
 Antetokounmpo and < 30}
```

Jika dokumen Anda memiliki objek bersarang dalam objek lain, Anda masih dapat mengambil data dengan menentukan semua level. Untuk melakukan ini, tempel yang berikut ini:

```
Top-Power-forwards.NBA players: {player-name:Lebron James}
```

Paket kustom untuk Amazon OpenSearch Service

OpenSearch Layanan Amazon memungkinkan Anda mengunggah file kamus khusus, seperti kata berhenti dan sinonim, dan juga menyediakan beberapa plugin opsional pra-paket yang dapat Anda kaitkan dengan domain Anda. Istilah umum untuk kedua jenis file ini adalah paket.

File kamus meningkatkan hasil pencarian Anda dengan mengatakan OpenSearch untuk mengabaikan kata-kata frekuensi tinggi tertentu atau untuk memperlakukan istilah seperti “puding beku,” “gelato,” dan “es krim” sebagai setara. Mereka juga dapat meningkatkan stemming, seperti di plugin Analisis Jepang (kuromoji).

Plugin opsional dapat memberikan fungsionalitas tambahan ke domain Anda. Misalnya, Anda dapat menggunakan plugin Amazon Personalize untuk memberi Anda hasil pencarian yang dipersonalisasi. Plugin opsional menggunakan jenis `ZIP-PLUGIN` paket. Untuk informasi selengkapnya tentang plugin opsional, lihatthe section called “Plugin berdasarkan versi mesin”.

Topik

- Persyaratan izin paket
- Mengunggah paket ke Amazon S3
- Mengimpor dan mengaitkan paket
- Menggunakan paket dengan OpenSearch
- Memperbarui paket
- Pembaruan indeks manual untuk kamus
- Memisahkan dan menghapus paket

Persyaratan izin paket

Pengguna tanpa akses administrator memerlukan tindakan AWS Identity and Access Management (IAM) tertentu untuk mengelola paket:

- `es:CreatePackage`- buat paket di Wilayah OpenSearch Layanan
- `es:DeletePackage`- menghapus paket dari Wilayah OpenSearch Layanan
- `es:AssociatePackage` - mengaitkan paket ke domain
- `es:DissociatePackage` - memisahkan paket dari domain

Anda juga perlu izin pada jalur bucket Amazon S3 atau objek di mana paket kustom berada.

Memberikan semua izin dalam IAM, tidak dalam kebijakan akses domain. Untuk informasi selengkapnya, lihat the section called “Manajemen Identitas dan Akses”.

Mengunggah paket ke Amazon S3

Bagian ini mencakup cara mengunggah paket kamus khusus, karena paket plugin opsional sudah diinstal sebelumnya. Sebelum Anda dapat mengaitkan kamus kustom dengan domain Anda, Anda harus mengunggahnya ke bucket Amazon S3. Untuk petunjuk, lihat Mengunggah objek di Panduan Pengguna Layanan Penyimpanan Sederhana Amazon. Plugin yang didukung tidak perlu diunggah.

Jika kamus berisi informasi sensitif, tentukan enkripsi sisi server dengan kunci yang dikelola S3 saat Anda mengunggahnya. OpenSearch Layanan tidak dapat mengakses file di S3 yang Anda lindungi menggunakan AWS KMS kunci.

Setelah Anda mengunggah file, catat jalur S3-nya. Formatnya jalurnya adalah `s3://`*`bucket-name`*`/`*`file-path`*`/`*`file-name`*.

Anda dapat menggunakan file sinonim berikut untuk tujuan pengujian. Simpan sebagai `synonyms.txt`.

```
danish, croissant, pastry
ice cream, gelato, frozen custard
sneaker, tennis shoe, running shoe
basketball shoe, hightop
```

Kamus tertentu, seperti kamus Hunspell, menggunakan banyak file dan memerlukan direktori mereka sendiri di sistem file. Pada saat ini, OpenSearch Layanan hanya mendukung kamus file tunggal.

Mengimpor dan mengaitkan paket

Konsol adalah cara paling sederhana untuk mengimpor kamus khusus ke OpenSearch Layanan. Saat Anda mengimpor kamus dari Amazon S3, OpenSearch Layanan menyimpan salinan paketnya sendiri dan secara otomatis mengenkripsi salinan itu menggunakan AES-256 dengan kunci yang dikelola Layanan. OpenSearch

Plugin opsional sudah diinstal sebelumnya di OpenSearch Layanan sehingga Anda tidak perlu mengunggahnya sendiri, tetapi Anda perlu mengaitkan plugin dengan domain. Plugin yang tersedia tercantum di layar Paket di konsol.

Impor dan kaitkan paket dengan domain dengan AWS Management Console

1. Di konsol OpenSearch Layanan Amazon, pilih Paket.
2. Pilih paket Impor.
3. Berikan kamus kustom nama deskriptif.
4. Berikan jalur S3 ke file, lalu pilih Kirim.
5. Kembali ke layar Paket.
6. Ketika status paket sudah Tersedia, pilih paket tersebut. Plugin opsional akan tersedia secara otomatis.
7. Pilih Kaitkan ke domain.
8. Pilih domain, lalu pilih Associate.
9. Di panel navigasi, pilih domain Anda dan buka tab Paket.
10. Jika paketnya adalah kamus khusus, perhatikan ID saat paket menjadi Tersedia. Gunakan `analyzers/id` sebagai jalur file dalam permintaan ke OpenSearch.

Sebagai alternatif, gunakan, SDK AWS CLI, atau API konfigurasi untuk mengimpor dan mengaitkan paket. Untuk informasi selengkapnya, lihat Referensi AWS CLI Perintah dan Referensi API Amazon OpenSearch Service.

Menggunakan paket dengan OpenSearch

Bagian ini mencakup cara menggunakan kedua jenis paket: kamus khusus dan plugin opsional.

Menggunakan kamus kustom

Setelah mengaitkan file dengan domain, Anda dapat menggunakannya dalam parameter seperti `synonyms_path`, `stopwords_path`, dan `user_dictionary` saat Anda membuat tokenizers dan filter token. Parameter yang tepat bervariasi menurut objek. Beberapa objek mendukung `synonyms_path` dan `stopwords_path`, namun `user_dictionary` eksklusif untuk plugin kuromoji.

Untuk plugin Analisis IK (Mandarin), Anda dapat mengunggah file kamus khusus sebagai paket khusus dan mengaitkannya ke domain, dan plugin secara otomatis mengambilnya tanpa memerlukan `user_dictionary` parameter. Jika file Anda adalah file sinonim, gunakan `synonyms_path` parameter.

Contoh berikut menambahkan file sinonim ke indeks baru:

```
PUT my-index
{
  "settings": {
    "index": {
      "analysis": {
        "analyzer": {
          "my_analyzer": {
            "type": "custom",
            "tokenizer": "standard",
            "filter": ["my_filter"]
          }
        },
        "filter": {
          "my_filter": {
            "type": "synonym",
            "synonyms_path": "analyzers/F111111111",
            "updateable": true
          }
        }
      }
    }
  },
  "mappings": {
    "properties": {
      "description": {
        "type": "text",
        "analyzer": "standard",
        "search_analyzer": "my_analyzer"
      }
    }
  }
}
```

Permintaan ini membuat penganalisis kustom untuk indeks yang menggunakan tokenizer standar dan filter token sinonim.

- Tokenizer memecah aliran karakter menjadi token (biasanya kata-kata) berdasarkan beberapa aturan. Contoh paling sederhana adalah tokenizer spasi, yang memecah karakter sebelumnya menjadi token setiap kali bertemu dengan karakter spasi. Contoh yang lebih kompleks adalah

tokenizer standar, yang menggunakan seperangkat aturan berbasis tata bahasa untuk bekerja di banyak bahasa.

- Filter Token menambahkan, memodifikasi, atau menghapus token. Misalnya, filter token sinonim menambahkan token ketika menemukan kata dalam daftar sinonim. Filter token berhenti menghapus token ketika menemukan kata dalam daftar kata berhenti.

Permintaan ini juga menambahkan bidang teks (`description`) ke pemetaan dan memberitahu OpenSearch untuk menggunakan penganalisis baru sebagai penganalisis pencariannya. Anda dapat melihat bahwa ia masih menggunakan penganalisis standar sebagai penganalisis indeksnya.

Akhirnya, perhatikan baris `"updateable": true` di filter token. Bidang ini hanya berlaku untuk penganalisis pencarian, bukan penganalisis indeks, dan sangat penting jika nanti Anda ingin memperbarui penganalisis pencarian secara otomatis.

Untuk tujuan pengujian, tambahkan beberapa dokumen ke indeks:

```
POST _bulk
{ "index": { "_index": "my-index", "_id": "1" } }
{ "description": "ice cream" }
{ "index": { "_index": "my-index", "_id": "2" } }
{ "description": "croissant" }
{ "index": { "_index": "my-index", "_id": "3" } }
{ "description": "tennis shoe" }
{ "index": { "_index": "my-index", "_id": "4" } }
{ "description": "hightop" }
```

Kemudian cari mereka menggunakan sinonim:

```
GET my-index/_search
{
  "query": {
    "match": {
      "description": "gelato"
    }
  }
}
```

Dalam hal ini, OpenSearch mengembalikan respons berikut:

```
{
```

```
  "hits": {
    "total": {
      "value": 1,
      "relation": "eq"
    },
    "max_score": 0.99463606,
    "hits": [{
      "_index": "my-index",
      "_type": "_doc",
      "_id": "1",
      "_score": 0.99463606,
      "_source": {
        "description": "ice cream"
      }
    }]
  }
}
```

> **Tip**
>
> File kamus menggunakan ruang timbunan Java sebanding dengan ukurannya. Sebagai contoh, file kamus 2 GiB mungkin mengonsumsi 2 GiB ruang tumpukan pada sebuah simpul. Jika Anda menggunakan file besar, pastikan bahwa simpul Anda memiliki cukup ruang tumpukan untuk mengakomodasinya. Pantau `JVMMemoryPressure`metrik, dan skalakan klaster Anda seperlunya.

Menggunakan plugin opsional

OpenSearch Layanan memungkinkan Anda mengaitkan OpenSearch plugin opsional yang sudah diinstal sebelumnya untuk digunakan dengan domain Anda. Paket plugin opsional kompatibel dengan OpenSearch versi tertentu, dan hanya dapat dikaitkan dengan domain dengan versi itu. Daftar paket yang tersedia untuk domain Anda mencakup semua plugin yang didukung yang kompatibel dengan versi domain Anda. Setelah Anda mengaitkan plugin ke domain, proses instalasi pada domain dimulai. Kemudian, Anda dapat mereferensikan dan menggunakan plugin saat Anda membuat permintaan ke OpenSearch Layanan.

Mengaitkan dan memisahkan plugin membutuhkan penerapan biru/hijau. Untuk informasi selengkapnya, lihat the section called “Perubahan yang biasanya menyebabkan penerapan biru/hijau”.

Plugin opsional termasuk penganalisis bahasa dan hasil pencarian yang disesuaikan. Misalnya, plugin Amazon Personalize Search Ranking menggunakan pembelajaran mesin untuk mempersonalisasi hasil penelusuran bagi pelanggan Anda. Untuk informasi selengkapnya tentang plugin ini, lihat Personalisasi hasil penelusuran dari OpenSearch. Untuk daftar semua plugin yang didukung, lihatthe section called “Plugin berdasarkan versi mesin”.

Plugin Sudachi

Untuk plugin Sudachi, ketika Anda mengasosiasikan kembali file kamus, itu tidak langsung mencerminkan domain. Kamus menyegarkan ketika penerapan biru/hijau berikutnya berjalan pada domain sebagai bagian dari perubahan konfigurasi atau pembaruan lainnya. Atau, Anda dapat membuat paket baru dengan data yang diperbarui, membuat indeks baru menggunakan paket baru ini, mengindeks ulang indeks yang ada ke indeks baru, dan kemudian menghapus indeks lama. Jika Anda lebih suka menggunakan pendekatan pengindeksan ulang, gunakan alias indeks sehingga tidak ada gangguan pada lalu lintas Anda.

Selain itu, plugin Sudachi hanya mendukung kamus Sudachi biner, yang dapat Anda unggah dengan operasi API. CreatePackage Untuk informasi tentang kamus sistem yang telah dibuat sebelumnya dan proses untuk menyusun kamus pengguna, lihat dokumentasi Sudachi.

Contoh berikut menunjukkan bagaimana menggunakan sistem dan kamus pengguna dengan tokenizer Sudachi. Anda harus mengunggah kamus ini sebagai paket khusus dengan tipe `TXT-DICTIONARY` dan memberikan ID paket mereka di pengaturan tambahan.

```
PUT sudachi_sample
{
  "settings": {
    "index": {
      "analysis": {
        "tokenizer": {
          "sudachi_tokenizer": {
            "type": "sudachi_tokenizer",
            "additional_settings": "{\"systemDict\": \"<system-dictionary-package-id>\",\"userDict\": [\"<user-dictionary-package-id>\"]}"
        }
        },
        "analyzer": {
          "sudachi_analyzer": {
            "filter": ["my_searchfilter"],
            "tokenizer": "sudachi_tokenizer",
            "type": "custom"
```

```
          }
        },
        "filter":{
          "my_searchfilter": {
            "type": "sudachi_split",
            "mode": "search"
          }
        }
      }
    }
  }
}
```

Memperbarui paket

Bagian ini hanya mencakup cara memperbarui paket kamus khusus, karena paket plugin opsional sudah diperbarui untuk Anda. Mengunggah versi baru kamus ke Amazon S3 tidak secara otomatis memperbarui paket di Layanan Amazon OpenSearch . OpenSearch Layanan menyimpan salinan file sendiri, jadi jika Anda mengunggah versi baru ke S3, Anda harus memperbaruinya secara manual.

Setiap domain yang terkait menyimpan salinan filenya sendiri. Agar perilaku pencarian dapat diprediksi, domain terus menggunakan versi paketnya saat ini hingga Anda memperbaruinya secara eksplisit. Untuk memperbarui paket kustom, ubah file Amazon S3 Control, perbarui paket di OpenSearch Layanan, lalu terapkan pembaruan.

Perbarui paket dengan AWS Management Console

1. Di konsol OpenSearch Layanan, pilih Paket.
2. Pilih paket dan Perbarui.
3. Sediakan jalur S3 ke file, dan kemudian pilih Perbarui paket.
4. Kembali ke layar Paket.
5. Ketika status paket berubah ke Tersedia, pilih paket tersebut. Kemudian pilih satu atau lebih domain terkait, Terapkan pembaruan, dan konfirmasi. Tunggu sampai status asosiasi berubah menjadi Aktif.
6. Langkah selanjutnya bervariasi tergantung pada bagaimana Anda mengonfigurasi indeks Anda:

   - Jika domain Anda berjalan OpenSearch atau Elasticsearch 7.8 atau yang lebih baru, dan hanya menggunakan penganalisis penelusuran dengan bidang yang dapat diperbarui disetel

ke true, Anda tidak perlu mengambil tindakan lebih lanjut. OpenSearch Layanan secara otomatis memperbarui indeks Anda menggunakan _plugins/_refresh_search_analyzers API.

- Jika domain Anda menjalankan Elasticsearch 7.7 atau yang lebih lama, menggunakan penganalisis indeks, atau tidak menggunakan bidang, lihat. `updateable` the section called "Pembaruan indeks manual untuk kamus"

Meskipun konsol adalah metode yang paling sederhana, Anda juga dapat menggunakan AWS CLI, SDK, atau API konfigurasi untuk memperbarui paket OpenSearch Layanan. Untuk informasi selengkapnya, lihat Referensi AWS CLI Perintah dan Referensi API Amazon OpenSearch Service.

Perbarui paket dengan AWS SDK

Alih-alih secara manual memperbarui paket di konsol, Anda dapat menggunakan SDK untuk mengotomatisasi proses pembaruan. Contoh skrip Python berikut mengunggah file paket baru ke Amazon S3, memperbarui paket di OpenSearch Layanan, dan menerapkan paket baru ke domain yang ditentukan. Setelah mengonfirmasi pembaruan berhasil, itu membuat panggilan sampel untuk OpenSearch menunjukkan sinonim baru telah diterapkan.

Anda harus memberikan nilai untuk `host`, `region`, `file_name`, `bucket_name`, `s3_key`, `package_id`, `domain_name`, dan `query`.

```
from requests_aws4auth import AWS4Auth
import boto3
import requests
import time
import json
import sys

host = ''  # The OpenSearch domain endpoint with https:// and a trailing slash. For
 example, https://my-test-domain.us-east-1.es.amazonaws.com/
region = ''  # For example, us-east-1
file_name = ''  # The path to the file to upload
bucket_name = ''  # The name of the S3 bucket to upload to
s3_key = ''  # The name of the S3 key (file name) to upload to
package_id = ''  # The unique identifier of the OpenSearch package to update
domain_name = ''  # The domain to associate the package with
query = ''  # A test query to confirm the package has been successfully updated

service = 'es'
credentials = boto3.Session().get_credentials()
client = boto3.client('opensearch')
```

```
awsauth = AWS4Auth(credentials.access_key, credentials.secret_key,
                   region, service, session_token=credentials.token)


def upload_to_s3(file_name, bucket_name, s3_key):
    """Uploads file to S3"""
    s3 = boto3.client('s3')
    try:
        s3.upload_file(file_name, bucket_name, s3_key)
        print('Upload successful')
        return True
    except FileNotFoundError:
        sys.exit('File not found. Make sure you specified the correct file path.')


def update_package(package_id, bucket_name, s3_key):
    """Updates the package in OpenSearch Service"""
    print(package_id, bucket_name, s3_key)
    response = client.update_package(
        PackageID=package_id,
        PackageSource={
            'S3BucketName': bucket_name,
            'S3Key': s3_key
        }
    )
    print(response)


def associate_package(package_id, domain_name):
    """Associates the package to the domain"""
    response = client.associate_package(
        PackageID=package_id, DomainName=domain_name)
    print(response)
    print('Associating...')


def wait_for_update(domain_name, package_id):
    """Waits for the package to be updated"""
    response = client.list_packages_for_domain(DomainName=domain_name)
    package_details = response['DomainPackageDetailsList']
    for package in package_details:
        if package['PackageID'] == package_id:
            status = package['DomainPackageStatus']
            if status == 'ACTIVE':
```

```
                print('Association successful.')
                return
            elif status == 'ASSOCIATION_FAILED':
                sys.exit('Association failed. Please try again.')
            else:
                time.sleep(10)  # Wait 10 seconds before rechecking the status
                wait_for_update(domain_name, package_id)


def sample_search(query):
    """Makes a sample search call to OpenSearch"""
    path = '_search'
    params = {'q': query}
    url = host + path
    response = requests.get(url, params=params, auth=awsauth)
    print('Searching for ' + '"' + query + '"')
    print(response.text)
```

> **Note**
>
> Jika Anda menerima kesalahan “paket tidak ditemukan” saat Anda menjalankan skrip menggunakan AWS CLI, kemungkinan besar Boto3 menggunakan Wilayah mana pun yang ditentukan dalam ~/.aws/config, yang bukan Wilayah bucket S3 tempat Anda berada. Jalankan `aws configure` dan tentukan Wilayah yang benar, atau tambahkan Region secara eksplisit ke klien:
>
> ```
> client = boto3.client('opensearch', region_name='us-east-1')
> ```

Pembaruan indeks manual untuk kamus

Pembaruan indeks manual hanya berlaku untuk kamus khusus, bukan plugin opsional. Untuk menggunakan kamus yang diperbarui, Anda harus memperbarui indeks secara manual jika memenuhi salah satu ketentuan berikut:

- Domain Anda menjalankan Elasticsearch 7.7 atau yang lebih lama.
- Anda menggunakan paket kustom sebagai indeks penganalisis.
- Anda menggunakan paket khusus sebagai penganalisis pencarian, tetapi tidak menyertakan bidang yang dapat diperbarui.

Untuk memperbarui penganalisis dengan file paket baru, Anda memiliki dua pilihan:

- Tutup dan buka indeks apa pun yang ingin Anda perbarui:

```
POST my-index/_close
POST my-index/_open
```

- Mengindeks ulang indeks. Pertama, buat indeks yang menggunakan file sinonim yang diperbarui (atau file yang sama sekali baru). Perhatikan bahwa hanya UTF-8 yang didukung.

```
PUT my-new-index
{
  "settings": {
    "index": {
      "analysis": {
        "analyzer": {
          "synonym_analyzer": {
            "type": "custom",
            "tokenizer": "standard",
            "filter": ["synonym_filter"]
          }
        },
        "filter": {
          "synonym_filter": {
            "type": "synonym",
            "synonyms_path": "analyzers/F222222222"
          }
        }
      }
    }
  },
  "mappings": {
    "properties": {
      "description": {
        "type": "text",
        "analyzer": "synonym_analyzer"
      }
    }
  }
}
```

Kemudian indeks ulang indeks lama ke indeks baru itu:

```
POST _reindex
{
  "source": {
    "index": "my-index"
  },
  "dest": {
    "index": "my-new-index"
  }
}
```

Jika Anda sering memperbarui penganalisis indeks, gunakan alias indeks untuk mempertahankan jalur yang konsisten ke indeks terbaru:

```
POST _aliases
{
  "actions": [
    {
      "remove": {
        "index": "my-index",
        "alias": "latest-index"
      }
    },
    {
      "add": {
        "index": "my-new-index",
        "alias": "latest-index"
      }
    }
  ]
}
```

Jika Anda tidak memerlukan indeks lama, hapus indeks tersebut:

```
DELETE my-index
```

Memisahkan dan menghapus paket

Memisahkan paket, apakah itu kamus khusus atau plugin opsional, dari domain berarti Anda tidak dapat lagi menggunakan paket itu saat membuat indeks baru. Setelah paket dipisahkan, indeks yang

ada yang menggunakan paket tidak dapat lagi menggunakannya. Anda harus menghapus paket dari indeks apa pun sebelum Anda dapat memisahkannya, jika tidak disosiasi gagal.

Konsol adalah cara paling sederhana untuk memisahkan paket dari domain dan menghapusnya dari OpenSearch Layanan. Menghapus paket dari OpenSearch Layanan tidak menghapusnya dari lokasi aslinya di Amazon S3.

Memisahkan paket dari domain dengan AWS Management Console

1. Masuk ke https://aws.amazon.com, dan kemudian pilih Masuk ke Konsol.
2. Di bawah Analytics, pilih OpenSearch Layanan Amazon.
3. Di panel navigasi, pilih domain Anda, lalu pilih tab Paket.
4. Pilih paket, Tindakan, dan kemudian pilih Dissociate. Konfirmasikan pilihan Anda.
5. Tunggu paket menghilang dari daftar. Anda mungkin harus me-refresh browser.
6. Jika Anda ingin menggunakan paket dengan domain lain, berhenti di sini. Untuk melanjutkan dengan menghapus paket (jika itu kamus kustom), pilih Paket di panel navigasi.
7. Pilih paket dan pilih Hapus.

Sebagai alternatif, gunakan, SDK AWS CLI, atau API konfigurasi untuk memisahkan dan menghapus paket. Untuk informasi selengkapnya, lihat Referensi AWS CLI Perintah dan Referensi API Amazon OpenSearch Service.

Menanyakan data Amazon OpenSearch Service Anda dengan SQL

Anda dapat menggunakan SQL untuk menanyakan OpenSearch Layanan Amazon Anda, daripada menggunakan DSL kueri berbasis JSONOpenSearch . Mengkuerikan dengan SQL berguna jika Anda sudah terbiasa dengan bahasa tersebut atau ingin mengintegrasikan domain Anda dengan aplikasi yang menggunakannya.

Gunakan tabel berikut untuk menemukan versi plugin SQL yang didukung oleh masing-masing OpenSearch dan versi Elasticsearch.

OpenSearch

OpenSearch versi	Versi plugin SQL	Fitur penting
2.11.0	2.11.0.0	Tambahkan dukungan untuk bahasa dan kueri PPL

OpenSearch versi	Versi plugin SQL	Fitur penting
2.9.0	2.9.0.0	Tambahkan konektor Spark, dan meja dukungan dan fungsi PromQL
2.7.0	2.7.0.0	Tambahkan `datasource` API
2.5.0	2.5.0.0	
2.3.0	2.3.0.0	Menambahkan `maketime` dan fungsi `makedate` datetime
1.3.0	1.3.0.0	Support ukuran batas kueri default, dan klausa IN untuk memilih dari dalam daftar nilai
1.2.0	1.2.0.0	Tambahkan protokol baru untuk format respons visualisasi
1.1.0	1.1.0.0	Mendukung fungsi pencocokan sebagai filter di SQL dan PPL
1.0.0	1.0.0.0	Support kueri aliran data

Buka Distro untuk Elasticsearch

Versi Elasticsearch	Versi plugin SQL	Fitur penting
7.10	1.13.0	NULL PERTAMA dan TERAKHIR untuk fungsi window, fungsi CAST(), perintah TAMPILKAN dan DESKRIPSIKAN
7.9	1.11.0	Menambahkan fungsi tanggal/waktu tambahan, URUTKAN BERDASARKAN kata kunci
7.8	1.9.0	
7.7	1.8.0	
7.3	1.3.0	Beberapa operator string dan nomor

Versi Elasticsearch	Versi plugin SQL	Fitur penting
7.1	1.1.0	

Dukungan SQL tersedia di domain yang berjalan OpenSearch atau Elasticsearch 6.5 atau lebih tinggi. Dokumentasi lengkap plugin SQL tersedia dalam OpenSearchdokumentasi.

Sampel panggilan

Untuk mengkueri data Anda dengan SQL, kirim permintaan HTTP ke `_sql` menggunakan format berikut:

```
POST domain-endpoint/_plugins/_sql
{
  "query": "SELECT * FROM my-index LIMIT 50"
}
```

> **Note**
>
> Jika domain Anda menjalankan Elasticsearch daripada OpenSearch, formatnya adalah. `_opendistro/_sql`

Catatan dan perbedaan

Panggilan ke `_plugins/_sql` menyertakan nama indeks dalam isi permintaan, sehingga mereka memiliki pertimbangan kebijakan akses yang sama seperti operasi bulk, mget, dan msearch. Seperti biasa, ikuti prinsip hak istimewa paling rendah ketika Anda memberikan izin ke operasi API.

Untuk pertimbangan keamanan terkait penggunaan SQL dengan kontrol akses berbutir halus, lihat. the section called “Kontrol akses detail”

Plugin OpenSearch SQL mencakup banyak pengaturan yang dapat disetel. Di OpenSearch Layanan, gunakan `_cluster/settings` jalur, bukan jalur pengaturan plugin (`_plugins/_query/settings`):

```
PUT _cluster/settings
```

```
{
  "transient" : {
    "plugins.sql.enabled" : true
  }
}
```

Untuk domain Elasticsearch lama, ganti dengan: `plugins` `opendistro`

```
PUT _cluster/settings
{
  "transient" : {
    "opendistro.sql.enabled" : true
  }
}
```

SQL Workbench

SQL Workbench adalah antarmuka pengguna OpenSearch Dasbor yang memungkinkan Anda menjalankan kueri SQL sesuai permintaan, menerjemahkan SQL ke dalam setara REST, dan melihat dan menyimpan hasil sebagai teks, JSON, JDBC, atau CSV. Untuk informasi selengkapnya, lihat Query Workbench.

SQL CLI

SQL CLI adalah aplikasi Python mandiri yang dapat Anda luncurkan dengan perintah `opensearchsql`. Untuk langkah-langkah menginstal, mengkonfigurasi, dan menggunakan, lihat SQL CLI.

Driver JDBC

Driver Java Database Connectivity (JDBC) memungkinkan Anda mengintegrasikan domain OpenSearch Layanan dengan aplikasi Business Intelligence (BI) favorit Anda. Untuk mengunduh driver, klik di sini. Untuk informasi lebih lanjut, lihat GitHub repositori.

Tabel berikut merangkum kompatibilitas versi untuk driver.

OpenSearch

OpenSearch versi	Versi driver JDBC
2.11	1.1.0.1

OpenSearch versi	Versi driver JDBC
2.9	1.1.0.1
2.7	1.1.0.1
2.5	1.1.0.1
2.3	1.1.0.1
1.3	1.1.0.1
1.2	1.1.0.1
1.1	1.1.0.1
1.0	1.1.0.1

Buka Distro untuk Elasticsearch

Versi Elasticsearch	Versi driver JDBC
7.10	1.13.0
7.9	1.11.0
7.8	1.9.0
7.7	1.8.0
7.4	1.4.0
7.1	1.0.0
6.8	0.9.0
6.7	0.9.0
6.5	0.9.0

Driver ODBC

Driver Open Database Connectivity (ODBC) adalah driver ODBC read-only untuk Windows dan macOS yang memungkinkan Anda menghubungkan intelijen bisnis dan aplikasi visualisasi data seperti Microsoft Excel ke plugin SQL.

Anda dapat mengunduh contoh file driver yang berfungsi di halaman OpenSearch artefak. Untuk informasi tentang menginstal driver, lihat repositori SQL pada. GitHub

Pencarian K-Nearest Neighbor (K-nN) di Amazon Service OpenSearch

Kependekan dari algoritma tetangga k-terdekat yang terkait, k-NN untuk Amazon OpenSearch Service memungkinkan Anda mencari titik dalam ruang vektor dan menemukan “tetangga terdekat” untuk titik-titik tersebut dengan jarak Euclidean atau kesamaan kosinus. Kasus penggunaan mencakup rekomendasi (misalnya, fitur "lagu lain yang mungkin Anda sukai" di aplikasi musik), pengenalan citra, dan deteksi penipuan.

Gunakan tabel berikut untuk menemukan versi plugin k-NN yang berjalan di domain Amazon OpenSearch Service Anda. Setiap versi plugin K-nn sesuai dengan versi OpenSearchatau Elasticsearch.

OpenSearch

OpenSearch versi	Versi plugin k-NN	Fitur penting
2.11	2.11.0.0	Ditambahkan dukungan untuk `ignore_unmapped` dalam kueri K-nn
2.9	2.9.0.0	Menerapkan vektor byte K-nN dan penyaringan yang efisien dengan mesin Faiss
2.7	2.7.0.0	
2.5	2.5.0.0	Diperpanjang SystemIndexPlugin untuk indeks sistem model K-NN, menambahkan ekstensi file khusus Lucene ke inti HybridFS

OpenSearch versi	Versi plugin k-NN	Fitur penting
2.3	2.3.0.0	
1.3	1.3.0.0	
1.2	1.2.0.0	Menambahkan dukungan untuk perpustakaan Faiss
1.1	1.1.0.0	
1.0	1.0.0.0	Mengganti nama REST API sambil mendukung kompatibi litas mundur, mengganti nama namespace dari ke `opendistro opensearch`

Elasticsearch

Versi elasticsearch	Versi plugin k-NN	Fitur penting
7.1	1.3.0.0	Jarak Euclidean
7.4	1.4.0.0	
7.7	1.8.0.0	Kesamaan kosinus
7.8	1.9.0.0	
7.9	1.11.0.0	API Warmup, penilaian khusus
7.10	1.13.0.0	Jarak Hamming, jarak L1 Norm, penulisan Painless

Dokumentasi lengkap untuk plugin K-NN tersedia dalam dokumentasi. OpenSearch Untuk informasi latar belakang tentang algoritme tetangga k-terdekat, lihat Wikipedia.

Memulai dengan k-NN

Untuk menggunakan k-NN, Anda harus membuat indeks dengan `index.knn` pengaturan dan menambahkan satu atau beberapa bidang tipe data `knn_vector`.

```
PUT my-index
{
  "settings": {
    "index.knn": true
  },
  "mappings": {
    "properties": {
      "my_vector1": {
        "type": "knn_vector",
        "dimension": 2
      },
      "my_vector2": {
        "type": "knn_vector",
        "dimension": 4
      }
    }
  }
}
```

Tipe data `knn_vector` mendukung daftar tunggal hingga 10.000 float, dengan jumlah float didefinisikan oleh parameter `dimension` yang diperlukan. Setelah Anda membuat indeks, tambahkan beberapa data untuk itu.

```
POST _bulk
{ "index": { "_index": "my-index", "_id": "1" } }
{ "my_vector1": [1.5, 2.5], "price": 12.2 }
{ "index": { "_index": "my-index", "_id": "2" } }
{ "my_vector1": [2.5, 3.5], "price": 7.1 }
{ "index": { "_index": "my-index", "_id": "3" } }
{ "my_vector1": [3.5, 4.5], "price": 12.9 }
{ "index": { "_index": "my-index", "_id": "4" } }
{ "my_vector1": [5.5, 6.5], "price": 1.2 }
{ "index": { "_index": "my-index", "_id": "5" } }
{ "my_vector1": [4.5, 5.5], "price": 3.7 }
{ "index": { "_index": "my-index", "_id": "6" } }
{ "my_vector2": [1.5, 5.5, 4.5, 6.4], "price": 10.3 }
{ "index": { "_index": "my-index", "_id": "7" } }
{ "my_vector2": [2.5, 3.5, 5.6, 6.7], "price": 5.5 }
{ "index": { "_index": "my-index", "_id": "8" } }
{ "my_vector2": [4.5, 5.5, 6.7, 3.7], "price": 4.4 }
{ "index": { "_index": "my-index", "_id": "9" } }
{ "my_vector2": [1.5, 5.5, 4.5, 6.4], "price": 8.9 }
```

Kemudian Anda dapat mencari data dengan menggunakan tipe kueri `knn`.

```
GET my-index/_search
{
  "size": 2,
  "query": {
    "knn": {
      "my_vector2": {
        "vector": [2, 3, 5, 6],
        "k": 2
      }
    }
  }
}
```

Dalam kasus ini, `k` adalah jumlah tetangga yang ingin Anda kueri agar kembali, tetapi Anda juga harus menyertakan opsi `size`. Jika tidak, Anda mendapatkan hasil `k` untuk setiap serpihan (dan setiap segmen) dan bukan hasil `k` untuk seluruh kueri. k-NN mendukung nilai `k` maksimal sebesar 10.000.

Jika Anda mencampur kueri `knn` dengan klausa lain, Anda mungkin menerima lebih sedikit dari hasil `k`. Dalam contoh ini, klausa `post_filter` mengurangi jumlah hasil dari 2 ke 1.

```
GET my-index/_search
{
  "size": 2,
  "query": {
    "knn": {
      "my_vector2": {
        "vector": [2, 3, 5, 6],
        "k": 2
      }
    }
  },
  "post_filter": {
    "range": {
      "price": {
        "gte": 6,
        "lte": 10
      }
    }
  }
```

```
}
```

Jika Anda perlu menangani sejumlah besar kueri sambil mempertahankan kinerja optimal, Anda dapat menggunakan `_msearch`API untuk membuat pencarian massal dengan JSON dan mengirim satu permintaan untuk melakukan beberapa pencarian:

```
GET _msearch
{ "index": "my-index"}
{ "query": { "knn": {"my_vector2":{"vector": [2, 3, 5, 6],"k":2 }} } }
{ "index": "my-index", "search_type": "dfs_query_then_fetch"}
{ "query": { "knn": {"my_vector1":{"vector": [2, 3],"k":2 }} } }
```

Video berikut menunjukkan cara mengatur pencarian vektor massal untuk kueri K-NN.

Perbedaan, penyetelan, dan batasan K-nn

OpenSearch memungkinkan Anda memodifikasi semua pengaturan K-nn menggunakan API. `_cluster/settings` Pada OpenSearch Layanan, Anda dapat mengubah semua pengaturan kecuali `knn.memory.circuit_breaker.enabled` dan`knn.circuit_breaker.triggered`. Statistik k-NN disertakan sebagai metrik Amazon CloudWatch .

Secara khusus, periksa `KNNGraphMemoryUsage` metrik pada setiap node data terhadap `knn.memory.circuit_breaker.limit` statistik dan RAM yang tersedia untuk jenis instance. OpenSearch Layanan menggunakan setengah dari RAM instance untuk heap Java (hingga ukuran heap 32 GiB). Secara default, k-NN menggunakan hingga 50% dari separuh yang tersisa, jadi tipe instans dengan 32 GiB RAM dapat menampung 8 GiB grafik (32 * 0.5 * 0.5). Performa dapat dirugikan jika penggunaan memori grafik melebihi nilai ini.

Anda tidak dapat memigrasikan indeks K-nn ke UltraWarmatau penyimpanan dingin jika indeks menggunakan perkiraan K-nn (). `"index.knn": true` Jika `index.knn` diatur ke `false` (persis K-nn), Anda masih dapat memindahkan indeks ke tingkatan penyimpanan lainnya.

Pencarian lintas-cluster di Layanan Amazon OpenSearch

Pencarian lintas klaster di Amazon OpenSearch Service memungkinkan Anda melakukan kueri dan agregasi di beberapa domain yang terhubung. Seringkali lebih masuk akal untuk menggunakan beberapa domain yang lebih kecil dan bukannya satu domain besar, terutama ketika Anda menjalankan berbagai jenis beban kerja.

Domain khusus beban kerja memungkinkan Anda untuk melakukan tugas berikut:

- Optimalkan setiap domain dengan memilih tipe instans untuk beban kerja tertentu.
- Menetapkan batas-batas kesalahan-isolasi di seluruh beban kerja. Ini berarti bahwa jika salah satu beban kerja Anda gagal, kesalahan yang terkandung dalam domain tertentu dan tidak mempengaruhi beban kerja Anda yang lain.
- Skala lebih mudah di seluruh domain.

Pencarian lintas cluster mendukung OpenSearch Dasbor, sehingga Anda dapat membuat visualisasi dan dasbor di semua domain Anda. Anda membayar biaya transfer AWS data standar untuk hasil pencarian yang ditransfer antar domain.

Topik

Batasan

Pencarian lintas klaster memiliki beberapa keterbatasan penting:

- Anda tidak dapat menghubungkan domain Elasticsearch ke domain. OpenSearch
- Anda tidak dapat terhubung ke klaster OpenSearch /Elasticsearch yang dikelola sendiri.
- Untuk menghubungkan domain di seluruh Wilayah, kedua domain harus berada di Elasticsearch 7.10 atau yang lebih baru atau. OpenSearch
- Sebuah domain dapat memiliki maksimum 20 koneksi keluar. Demikian pula, domain dapat memiliki maksimum 20 koneksi masuk. Dengan kata lain, satu domain dapat terhubung ke maksimum 20 domain lainnya.
- Domain sumber harus pada versi yang sama atau lebih tinggi dari domain tujuan. Jika Anda mengatur koneksi dua arah antara dua domain dan Anda ingin memutakhirkan satu atau keduanya, Anda harus terlebih dahulu menghapus salah satu koneksi.
- Anda tidak dapat menggunakan kamus kustom atau SQL dengan pencarian lintas-klaster.

- Anda tidak dapat menggunakan AWS CloudFormation untuk menghubungkan domain.
- Anda tidak dapat menggunakan pencarian lintas klaster pada instance M3 atau burstable (T2 dan T3).

Prasyarat pencarian lintas klaster

Sebelum Anda mengatur pencarian lintas-klaster, pastikan bahwa domain Anda memenuhi persyaratan berikut:

- Dua OpenSearch domain, atau domain Elasticsearch pada versi 6.7 atau yang lebih baru
- Kontrol akses detail diaktifkan
- ode-to-node Enkripsi N diaktifkan

Penentuan harga pencarian lintas klaster

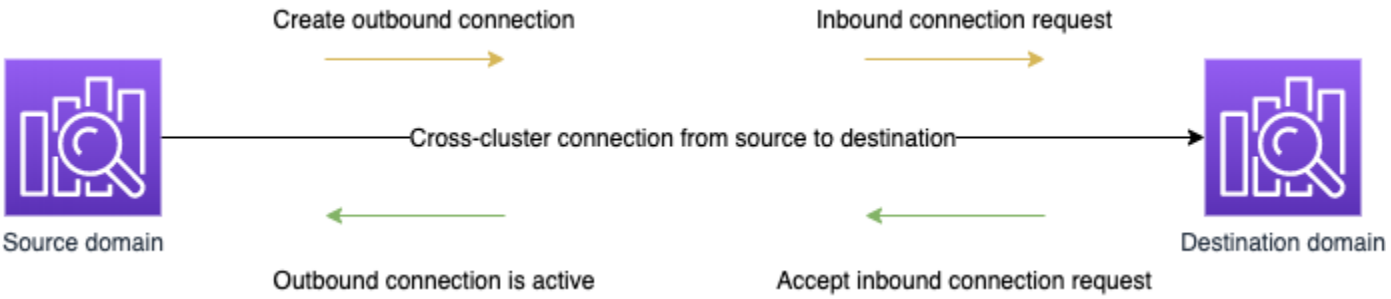
Tidak ada biaya tambahan untuk mencari di seluruh domain.

Menyiapkan koneksi

Domain “sumber” mengacu pada domain yang berasal dari permintaan pencarian lintas klaster dari . Dengan kata lain, domain sumber adalah domain tempat Anda mengirim permintaan pencarian awal.

Domain "tujuan" adalah domain yang ditanyakan oleh domain sumber.

Koneksi lintas klaster searah dari sumber ke domain tujuan. Ini berarti bahwa domain tujuan tidak dapat membuat kueri domain sumber. Namun, Anda dapat mengatur koneksi lain ke arah yang berlawanan.



Sumber domain membuat koneksi “keluar” ke domain tujuan. Domain tujuan menerima permintaan koneksi “masuk” dari domain sumber.

Untuk mengatur koneksi

1. Di dasbor domain Anda, pilih domain dan buka tab Koneksi.
2. Di bagian Koneksi keluar, pilih Permintaan.
3. Untuk alias Koneksi, masukkan nama untuk koneksi Anda.
4. Pilih antara menghubungkan ke domain di wilayah Anda Akun AWS atau di akun atau Wilayah lain.

   - Untuk terhubung ke cluster di Region Akun AWS dan Anda, pilih domain dari menu dropdown dan pilih Request.
   - Untuk terhubung ke cluster di wilayah lain Akun AWS atau, pilih ARN dari domain jarak jauh dan pilih Permintaan. Untuk menghubungkan domain di seluruh Wilayah, kedua domain harus menjalankan Elasticsearch versi 7.10 atau yang lebih baru atau. OpenSearch
5. Untuk melewati klaster yang tidak tersedia untuk kueri klaster, pilih Lewati tidak tersedia. Pengaturan ini memastikan bahwa kueri lintas klaster Anda mengembalikan sebagian hasil meskipun gagal pada satu atau beberapa klaster jarak jauh.
6. Pencarian lintas-cluster pertama memvalidasi permintaan koneksi untuk memastikan prasyarat terpenuhi. Jika domain ditemukan tidak kompatibel, permintaan koneksi memasuki status. `Validation failed`
7. Setelah permintaan koneksi berhasil divalidasi, lalu dikirim ke domain tujuan, di mana perlu disetujui. Sampai persetujuan ini terjadi, koneksi tetap dalam `Pending acceptance` keadaan. Ketika permintaan koneksi diterima di domain tujuan, status berubah `Active` dan domain tujuan menjadi tersedia untuk kueri.

   - Halaman domain menunjukkan kesehatan domain secara keseluruhan dan detail kesehatan instans domain tujuan Anda. Hanya pemilik domain yang memiliki fleksibilitas untuk membuat, melihat, menghapus, dan memantau koneksi ke atau dari domain mereka.

Setelah koneksi dibuat, lalu lintas yang mengalir di antara simpul dari domain yang terhubung dienkripsi. Jika Anda menghubungkan domain VPC ke domain non-VPC dan domain non-VPC adalah titik akhir publik yang dapat menerima lalu lintas dari internet, lalu lintas antar klaster antara domain masih dienkripsi dan aman.

Menghapus koneksi

Menghapus koneksi menghentikan setiap operasi lintas-klaster pada indeksnya.

1. Di dasbor domain Anda, buka tab Koneksi.
2. Pilih koneksi domain yang ingin Anda hapus dan pilih Hapus, lalu konfirmasikan penghapusan.

Anda dapat melakukan langkah-langkah ini pada domain sumber atau tujuan untuk menghapus koneksi. Setelah Anda menghapus koneksi, itu masih terlihat dengan `Deleted` status untuk jangka waktu 15 hari.

Anda tidak dapat menghapus domain dengan koneksi lintas klaster yang aktif. Untuk menghapus domain, pertama-tama hapus semua koneksi masuk dan keluar dari domain tersebut. Ini memastikan Anda memperhitungkan pengguna domain lintas cluster sebelum menghapus domain.

Menyiapkan keamanan dan sampel panduan

1. Anda mengirim permintaan pencarian lintas klaster untuk domain sumber.
2. Domain sumber mengevaluasi permintaan tersebut terhadap kebijakan akses domainnya. Karena pencarian lintas klaster memerlukan kontrol akses detail, kami merekomendasikan kebijakan akses terbuka pada domain sumber.

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Effect": "Allow",
      "Principal": {
        "AWS": [
          "*"
        ]
      },
      "Action": [
        "es:ESHttp*"
      ],
      "Resource": "arn:aws:es:region:account:domain/src-domain/*"
    }
  ]
}
```

> **Note**
>
> Jika Anda menyertakan indeks jarak jauh di jalur, Anda harus mengkodekan URL URI di domain ARN. Misalnya, gunakan `arn:aws:es:us-east-1:123456789012:domain/my-domain/local_index,dst%3Aremote_index` bukan`arn:aws:es:us-east-1:123456789012:domain/my-domain/local_index,dst:remote_index`.

Jika Anda memilih untuk menggunakan kebijakan akses terbatas selain kontrol akses detail, minimal kebijakan Anda harus mengizinkan akses ke `es:ESHttpGet`.

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Effect": "Allow",
      "Principal": {
        "AWS": [
          "arn:aws:iam::123456789012:user/test-user"
        ]
      },
      "Action": "es:ESHttpGet",
      "Resource": "arn:aws:es:region:account:domain/src-domain/*"
    }
  ]
}
```

3. Kontrol akses detail pada domain sumber mengevaluasi permintaan:

   - Apakah permintaan ditandatangani dengan kredensial dasar IAM atau HTTP yang valid?
   - Jika demikian, apakah pengguna memiliki izin untuk melakukan pencarian dan mengakses data?

   Jika permintaan hanya mencari data pada domain tujuan (misalnya, `dest-alias:dest-index/_search`), Anda hanya memerlukan izin pada domain tujuan.

   Jika permintaan mencari data di kedua domain (misalnya, `source-index,dest-alias:dest-index/_search`), Anda memerlukan izin pada kedua domain.

Dalam kontrol akses detail, pengguna harus memiliki izin `indices:admin/shards/search_shards` selain `read` standar atau izin `search` untuk indeks yang relevan.

4. Sumber domain melewati permintaan ke domain tujuan. Domain tujuan mengevaluasi permintaan ini terhadap kebijakan akses domainnya. Anda harus menyertakan izin `es:ESCrossClusterGet` pada domain tujuan:

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Effect": "Allow",
      "Principal": {
        "AWS": "*"
      },
      "Action": "es:ESCrossClusterGet",
      "Resource": "arn:aws:es:region:account:domain/dst-domain"
    }
  ]
}
```

Pastikan bahwa izin `es:ESCrossClusterGet` diterapkan untuk `/dst-domain` dan bukan `/dst-domain/*`.

Namun, kebijakan minimum ini hanya memungkinkan pencarian lintas klaster. Untuk melakukan operasi lain, seperti mengindeks dokumen dan melakukan pencarian standar, Anda memerlukan izin tambahan. Kami merekomendasikan kebijakan berikut pada domain tujuan:

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Effect": "Allow",
      "Principal": {
        "AWS": [
          "*"
        ]
      },
      "Action": [
        "es:ESHttp*"
      ],
```

```
      "Resource": "arn:aws:es:region:account:domain/dst-domain/*"
    },
    {
      "Effect": "Allow",
      "Principal": {
        "AWS": "*"
      },
      "Action": "es:ESCrossClusterGet",
      "Resource": "arn:aws:es:region:account:domain/dst-domain"
    }
  ]
}
```

> **Note**
>
> Semua permintaan pencarian lintas cluster antar domain dienkripsi dalam perjalanan secara default sebagai bagian dari enkripsi. node-to-node

5. Domain tujuan melakukan pencarian dan mengembalikan hasil ke domain sumber.
6. Sumber domain menggabungkan hasil sendiri (jika ada) dengan hasil dari domain tujuan dan mengembalikannya kepada Anda.
7. Kami merekomendasikan Postman untuk permintaan pengujian:

   - Pada domain tujuan, indeks dokumen:

   ```
   POST https://dst-domain.us-east-1.es.amazonaws.com/books/_doc/1

   {
     "Dracula": "Bram Stoker"
   }
   ```

   - Untuk melakukan kueri indeks ini dari domain sumber, sertakan alias koneksi domain tujuan dalam kueri.

   ```
   GET https://src-domain.us-east-1.es.amazonaws.com/<connection_alias>:books/
   _search

   {
       ...
     "hits": [
       {
   ```

```
      "_index": "source-destination:books",
      "_type": "_doc",
      "_id": "1",
      "_score": 1,
      "_source": {
        "Dracula": "Bram Stoker"
      }
    }
  ]
}
```

Anda dapat menemukan alias koneksi di tab Connections di dasbor domain Anda.

- Jika Anda mengatur koneksi antara `domain-a -> domain-b` dengan alias `cluster_b` dan `domain-a -> domain-c` dengan alias koneksi `cluster_c`, pencarian `domain-a`, `domain-b`, dan `domain-c` sebagai berikut:

```
GET https://src-domain.us-east-1.es.amazonaws.com/
local_index,cluster_b:b_index,cluster_c:c_index/_search
{
  "query": {
    "match": {
      "user": "domino"
    }
  }
}
```

Respons

```
{
  "took": 150,
  "timed_out": false,
  "_shards": {
    "total": 3,
    "successful": 3,
    "failed": 0,
    "skipped": 0
  },
  "_clusters": {
    "total": 3,
    "successful": 3,
    "skipped": 0
```

```
  },
  "hits": {
    "total": 3,
    "max_score": 1,
    "hits": [
      {
        "_index": "local_index",
        "_type": "_doc",
        "_id": "0",
        "_score": 1,
        "_source": {
          "user": "domino",
          "message": "Lets unite the new mutants",
          "likes": 0
        }
      },
      {
        "_index": "cluster_b:b_index",
        "_type": "_doc",
        "_id": "0",
        "_score": 2,
        "_source": {
          "user": "domino",
          "message": "I'm different",
          "likes": 0
        }
      },
      {
        "_index": "cluster_c:c_index",
        "_type": "_doc",
        "_id": "0",
        "_score": 3,
        "_source": {
          "user": "domino",
          "message": "So am I",
          "likes": 0
        }
      }
    ]
  }
}
```

Jika Anda tidak memilih untuk melewati kluster yang tidak tersedia dalam pengaturan koneksi, semua kluster tujuan yang Anda cari harus tersedia agar permintaan pencarian Anda berjalan dengan sukses. Jika tidak, seluruh permintaan gagal—bahkan jika salah satu domain tidak tersedia, tidak ada hasil pencarian yang dikembalikan.

OpenSearch Dasbor

Anda dapat memvisualisasikan data dari beberapa domain yang terhubung dengan cara yang sama seperti dari satu domain, kecuali bahwa Anda harus mengakses indeks jarak jauh menggunakan. `connection-alias:index` Jadi, pola indeks Anda harus cocok dengan `connection-alias:index`.

Belajar Peringkat untuk OpenSearch Layanan Amazon

OpenSearch menggunakan kerangka peringkat probabilistik yang disebut BM-25 untuk menghitung skor relevansi. Jika kata kunci khas muncul lebih sering dalam dokumen, BM-25 memberikan skor relevansi yang lebih tinggi untuk dokumen tersebut. Akan tetapi, kerangka kerja ini tidak memperhitungkan perilaku pengguna akun seperti data klik-tayang, yang dapat lebih meningkatkan relevansi.

Learning to Rank adalah plugin open-source yang memungkinkan Anda menggunakan pembelajaran mesin dan data perilaku untuk menyesuaikan relevansi dokumen. Ini menggunakan model dari perpustakaan XGBoost dan Ranklib untuk mencetak ulang hasil pencarian. Plugin Elasticsearch LTR awalnya dikembangkan oleh OpenSource Connections, dengan kontribusi signifikan oleh Wikimedia Foundation, Snagajob Engineering, Bonsai, dan Yelp Engineering. OpenSearch Versi plugin berasal dari plugin Elasticsearch LTR. Dokumentasi lengkap, termasuk langkah-langkah rinci dan deskripsi API, tersedia dalam dokumentasi Learning to Rank.

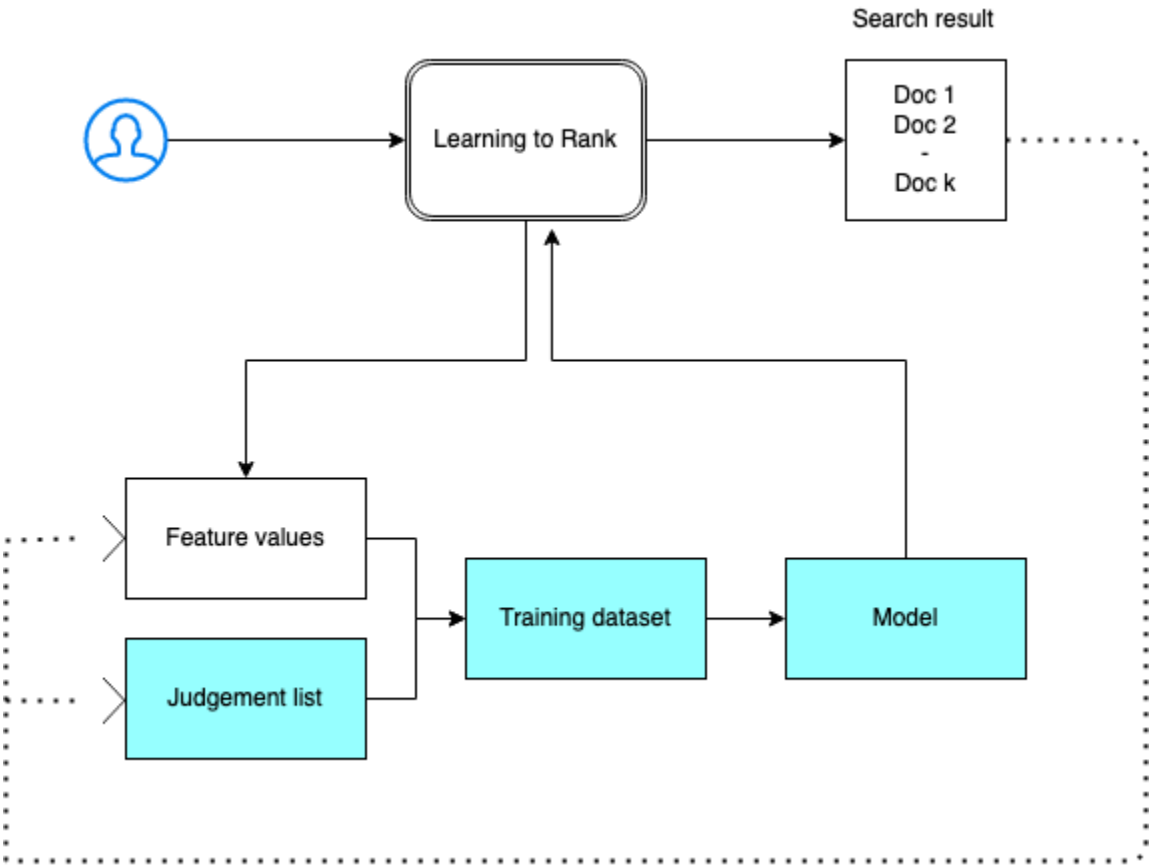
Belajar Peringkat membutuhkan OpenSearch atau Elasticsearch 7.7 atau yang lebih baru.

> **Note**
>
> Untuk menggunakan plugin Learning to Rank, Anda harus memiliki izin admin lengkap. Untuk mempelajari selengkapnya, lihat the section called “Mengubah pengguna utama”.

Topik

Memulai dengan Learning to Rank

Anda perlu memberikan daftar penilaian, menyiapkan kumpulan data pelatihan, dan melatih model di luar OpenSearch Layanan Amazon. Bagian berwarna biru terjadi di luar OpenSearch Layanan:



Langkah 1: Menginisialisasi plugin

Untuk menginisialisasi plugin Learning to Rank, kirim permintaan berikut ke domain OpenSearch Layanan Anda:

```
PUT _ltr
```

```
{
  "acknowledged" : true,
  "shards_acknowledged" : true,
  "index" : ".ltrstore"
```

```
}
```

Perintah ini membuat indeks `.ltrstore` tersembunyi yang menyimpan informasi metadata seperti set dan model fitur.

Langkah 2: Buat daftar penilaian

> **Note**
>
> Anda harus melakukan langkah ini di luar OpenSearch Layanan.

Daftar penilaian adalah kumpulan contoh yang dipelajari oleh model machine learning. Daftar penilaian Anda harus mencakup kata kunci yang penting bagi Anda dan satu set dokumen yang telah dinilai untuk setiap kata kunci.

Dalam contoh ini, kita memiliki daftar penilaian untuk set data film. Nilai 4 menunjukkan kecocokan yang sempurna. Nilai 0 menunjukkan pertandingan terburuk.

Nilai	Kata Kunci	ID Dokumen	Nama film
4	rambo	7555	Rambo
3	rambo	1370	Rambo III
3	rambo	1369	Rambo: First Blood Part II
3	rambo	1368	First Blood

Siapkan daftar penilaian Anda dalam format berikut:

```
4 qid:1 # 7555 Rambo
3 qid:1 # 1370 Rambo III
3 qid:1 # 1369 Rambo: First Blood Part II
3 qid:1 # 1368 First Blood

where qid:1 represents "rambo"
```

Untuk contoh daftar penilaian yang lebih lengkap, lihat penilaian film.

Anda dapat membuat daftar penilaian ini secara manual dengan bantuan anotator manusia atau menyimpulkan secara terprogram dari data analitik.

Langkah 3: Bangun satu set fitur

Fitur adalah bidang yang sesuai dengan relevansi dokumen—misalnya, `title`, `overview`, `popularity score` (jumlah tampilan), dan sebagainya.

Bangun satu set fitur dengan template Mustache untuk setiap fitur. Untuk informasi selengkapnya tentang fitur, lihat Bekerja dengan Fitur.

Dalam contoh ini, kita membangun fitur `movie_features` yang ditetapkan dengan bidang `title` dan `overview`:

```
POST _ltr/_featureset/movie_features
{
  "featureset" : {
      "name" : "movie_features",
      "features" : [
        {
          "name" : "1",
          "params" : [
            "keywords"
          ],
          "template_language" : "mustache",
          "template" : {
            "match" : {
              "title" : "{{keywords}}"
            }
          }
        },
        {
          "name" : "2",
          "params" : [
            "keywords"
          ],
          "template_language" : "mustache",
          "template" : {
            "match" : {
              "overview" : "{{keywords}}"
            }
```

```
          }
        }
      ]
    }
}
```

Jika mengajukan kueri untuk indeks `.ltrstore` asli, Anda mendapatkan kembali set fitur Anda:

```
GET _ltr/_featureset
```

Langkah 4: Log nilai fitur

Nilai fitur adalah skor relevansi yang dihitung oleh BM-25 untuk setiap fitur.

Menggabungkan set fitur dan daftar penilaian untuk mencatat log nilai fitur. Untuk informasi selengkapnya tentang fitur pencatatan log, lihat Skor Fitur Pencatatan Log.

Dalam contoh ini, kueri `bool` mengambil dokumen yang sudah dinilai dengan filter, kemudian memilih set fitur dengan kueri `sltr`. Kueri `ltr_log` menggabungkan dokumen dan fitur untuk mencatat log nilai fitur yang sesuai:

```
POST tmdb/_search
{
  "_source": {
    "includes": [
      "title",
      "overview"
    ]
  },
  "query": {
    "bool": {
      "filter": [
        {
          "terms": {
            "_id": [
              "7555",
              "1370",
              "1369",
              "1368"
            ]
          }
        },
```

```
        {
          "sltr": {
            "_name": "logged_featureset",
            "featureset": "movie_features",
            "params": {
              "keywords": "rambo"
            }
          }
        }
      ]
    }
  },
  "ext": {
    "ltr_log": {
      "log_specs": {
        "name": "log_entry1",
        "named_query": "logged_featureset"
      }
    }
  }
}
```

Respons sampel mungkin terlihat seperti berikut ini:

```
{
  "took" : 7,
  "timed_out" : false,
  "_shards" : {
    "total" : 1,
    "successful" : 1,
    "skipped" : 0,
    "failed" : 0
  },
  "hits" : {
    "total" : {
      "value" : 4,
      "relation" : "eq"
    },
    "max_score" : 0.0,
    "hits" : [
      {
        "_index" : "tmdb",
        "_type" : "movie",
```

```
        "_id" : "1368",
        "_score" : 0.0,
        "_source" : {
          "overview" : "When former Green Beret John Rambo is harassed by local law
enforcement and arrested for vagrancy, the Vietnam vet snaps, runs for the hills and
rat-a-tat-tats his way into the action-movie hall of fame. Hounded by a relentless
sheriff, Rambo employs heavy-handed guerilla tactics to shake the cops off his tail.",
          "title" : "First Blood"
        },
        "fields" : {
          "_ltrlog" : [
            {
              "log_entry1" : [
                {
                  "name" : "1"
                },
                {
                  "name" : "2",
                  "value" : 10.558305
                }
              ]
            }
          ]
        },
        "matched_queries" : [
          "logged_featureset"
        ]
      },
      {
        "_index" : "tmdb",
        "_type" : "movie",
        "_id" : "7555",
        "_score" : 0.0,
        "_source" : {
          "overview" : "When governments fail to act on behalf of captive missionaries,
ex-Green Beret John James Rambo sets aside his peaceful existence along the Salween
River in a war-torn region of Thailand to take action.  Although he's still haunted
by violent memories of his time as a U.S. soldier during the Vietnam War, Rambo can
hardly turn his back on the aid workers who so desperately need his help.",
          "title" : "Rambo"
        },
        "fields" : {
          "_ltrlog" : [
            {
```

```
            "log_entry1" : [
              {
                "name" : "1",
                "value" : 11.2569065
              },
              {
                "name" : "2",
                "value" : 9.936821
              }
            ]
          }
        ]
      },
      "matched_queries" : [
        "logged_featureset"
      ]
    },
    {
      "_index" : "tmdb",
      "_type" : "movie",
      "_id" : "1369",
      "_score" : 0.0,
      "_source" : {
        "overview" : "Col. Troutman recruits ex-Green Beret John Rambo for a highly
secret and dangerous mission. Teamed with Co Bao, Rambo goes deep into Vietnam to
rescue POWs. Deserted by his own team, he's left in a hostile jungle to fight for his
life, avenge the death of a woman and bring corrupt officials to justice.",
        "title" : "Rambo: First Blood Part II"
      },
      "fields" : {
        "_ltrlog" : [
          {
            "log_entry1" : [
              {
                "name" : "1",
                "value" : 6.334839
              },
              {
                "name" : "2",
                "value" : 10.558305
              }
            ]
          }
        ]
```

```
        },
        "matched_queries" : [
          "logged_featureset"
        ]
      },
      {
        "_index" : "tmdb",
        "_type" : "movie",
        "_id" : "1370",
        "_score" : 0.0,
        "_source" : {
          "overview" : "Combat has taken its toll on Rambo, but he's finally begun to
 find inner peace in a monastery. When Rambo's friend and mentor Col. Trautman asks for
 his help on a top secret mission to Afghanistan, Rambo declines but must reconsider
 when Trautman is captured.",
          "title" : "Rambo III"
        },
        "fields" : {
          "_ltrlog" : [
            {
              "log_entry1" : [
                {
                  "name" : "1",
                  "value" : 9.425955
                },
                {
                  "name" : "2",
                  "value" : 11.262714
                }
              ]
            }
          ]
        },
        "matched_queries" : [
          "logged_featureset"
        ]
      }
    ]
  }
}
```

Pada contoh sebelumnya, fitur pertama tidak memiliki nilai fitur karena kata kunci “rambo” tidak muncul di bidang judul dokumen dengan ID sama dengan 1368. Ini adalah nilai fitur yang hilang dalam data pelatihan.

Langkah 5: Buat set data pelatihan

> **Note**
>
> Anda harus melakukan langkah ini di luar OpenSearch Layanan.

Langkah selanjutnya adalah menggabungkan daftar penilaian dan nilai fitur untuk membuat set data pelatihan. Jika daftar penilaian asli Anda terlihat seperti ini:

```
4 qid:1 # 7555 Rambo
3 qid:1 # 1370 Rambo III
3 qid:1 # 1369 Rambo: First Blood Part II
3 qid:1 # 1368 First Blood
```

Lakukan konversi ke dalam set data pelatihan akhir, yang terlihat seperti ini:

```
4 qid:1 1:12.318474 2:10.573917 # 7555 rambo
3 qid:1 1:10.357875 2:11.950391 # 1370 rambo
3 qid:1 1:7.010513 2:11.220095 # 1369 rambo
3 qid:1 1:0.0 2:11.220095 # 1368 rambo
```

Anda dapat melakukan langkah ini secara manual atau menulis program untuk mengotomatisasinya.

Langkah 6: Pilih algoritme dan membangun model

> **Note**
>
> Anda harus melakukan langkah ini di luar OpenSearch Layanan.

Dengan set data pelatihan di tempat, langkah selanjutnya adalah menggunakan perpustakaan XGBoost atau Ranklib untuk membangun sebuah model. Perpustakaan XGBoost dan Ranklib memungkinkan Anda membangun model populer seperti LambdamArt, Random Forest, dan sebagainya.

Untuk langkah-langkah menggunakan XGBoost dan Ranklib untuk membangun model, lihat XGBoost dan dokumentasi, masing-masing. RankLib Untuk menggunakan Amazon SageMaker untuk membangun model XGBoost, lihat Algoritma XGBoost.

Langkah 7: Men-deploy model

Setelah Anda membuat model, deploy-kan model tersebut ke plugin Learning to Rank. Untuk informasi selengkapnya tentang proses deploy model, lihat Mengunggah Model Terlatih.

Dalam contoh ini, kami membangun model `my_ranklib_model` dengan menggunakan pustaka Ranklib:

```
POST _ltr/_featureset/movie_features/_createmodel?pretty
{
  "model": {
    "name": "my_ranklib_model",
    "model": {
      "type": "model/ranklib",
      "definition": """## LambdaMART
## No. of trees = 10
## No. of leaves = 10
## No. of threshold candidates = 256
## Learning rate = 0.1
## Stop early = 100

<ensemble>
   <tree id="1" weight="0.1">
      <split>
         <feature>1</feature>
         <threshold>10.357875</threshold>
         <split pos="left">
            <feature>1</feature>
            <threshold>0.0</threshold>
            <split pos="left">
               <output>-2.0</output>
            </split>
            <split pos="right">
               <feature>1</feature>
               <threshold>7.010513</threshold>
               <split pos="left">
                  <output>-2.0</output>
               </split>
               <split pos="right">
```

```
                <output>-2.0</output>
             </split>
          </split>
       </split>
       <split pos="right">
          <output>2.0</output>
       </split>
    </split>
 </tree>
 <tree id="2" weight="0.1">
    <split>
       <feature>1</feature>
       <threshold>10.357875</threshold>
       <split pos="left">
          <feature>1</feature>
          <threshold>0.0</threshold>
          <split pos="left">
             <output>-1.67031991481781</output>
          </split>
          <split pos="right">
             <feature>1</feature>
             <threshold>7.010513</threshold>
             <split pos="left">
                <output>-1.67031991481781</output>
             </split>
             <split pos="right">
                <output>-1.6703200340270996</output>
             </split>
          </split>
       </split>
       <split pos="right">
          <output>1.6703201532363892</output>
       </split>
    </split>
 </tree>
 <tree id="3" weight="0.1">
    <split>
       <feature>2</feature>
       <threshold>10.573917</threshold>
       <split pos="left">
          <output>1.479954481124878</output>
       </split>
       <split pos="right">
          <feature>1</feature>
```

```
            <threshold>7.010513</threshold>
            <split pos="left">
               <feature>1</feature>
               <threshold>0.0</threshold>
               <split pos="left">
                  <output>-1.4799546003341675</output>
               </split>
               <split pos="right">
                  <output>-1.479954481124878</output>
               </split>
            </split>
            <split pos="right">
               <output>-1.479954481124878</output>
            </split>
         </split>
      </split>
   </tree>
   <tree id="4" weight="0.1">
      <split>
         <feature>1</feature>
         <threshold>10.357875</threshold>
         <split pos="left">
            <feature>1</feature>
            <threshold>0.0</threshold>
            <split pos="left">
               <output>-1.3569872379302979</output>
            </split>
            <split pos="right">
               <feature>1</feature>
               <threshold>7.010513</threshold>
               <split pos="left">
                  <output>-1.3569872379302979</output>
               </split>
               <split pos="right">
                  <output>-1.3569872379302979</output>
               </split>
            </split>
         </split>
         <split pos="right">
            <output>1.3569873571395874</output>
         </split>
      </split>
   </tree>
   <tree id="5" weight="0.1">
```

```
      <split>
         <feature>1</feature>
         <threshold>10.357875</threshold>
         <split pos="left">
            <feature>1</feature>
            <threshold>0.0</threshold>
            <split pos="left">
               <output>-1.2721362113952637</output>
            </split>
            <split pos="right">
               <feature>1</feature>
               <threshold>7.010513</threshold>
               <split pos="left">
                  <output>-1.2721363306045532</output>
               </split>
               <split pos="right">
                  <output>-1.2721363306045532</output>
               </split>
            </split>
         </split>
         <split pos="right">
            <output>1.2721362113952637</output>
         </split>
      </split>
   </tree>
   <tree id="6" weight="0.1">
      <split>
         <feature>1</feature>
         <threshold>10.357875</threshold>
         <split pos="left">
            <feature>1</feature>
            <threshold>7.010513</threshold>
            <split pos="left">
               <feature>1</feature>
               <threshold>0.0</threshold>
               <split pos="left">
                  <output>-1.2110036611557007</output>
               </split>
               <split pos="right">
                  <output>-1.2110036611557007</output>
               </split>
            </split>
            <split pos="right">
               <output>-1.2110037803649902</output>
```

```
            </split>
         </split>
         <split pos="right">
            <output>1.2110037803649902</output>
         </split>
      </split>
   </tree>
   <tree id="7" weight="0.1">
      <split>
         <feature>1</feature>
         <threshold>10.357875</threshold>
         <split pos="left">
            <feature>1</feature>
            <threshold>7.010513</threshold>
            <split pos="left">
               <feature>1</feature>
               <threshold>0.0</threshold>
               <split pos="left">
                  <output>-1.165616512298584</output>
               </split>
               <split pos="right">
                  <output>-1.165616512298584</output>
               </split>
            </split>
            <split pos="right">
               <output>-1.165616512298584</output>
            </split>
         </split>
         <split pos="right">
            <output>1.165616512298584</output>
         </split>
      </split>
   </tree>
   <tree id="8" weight="0.1">
      <split>
         <feature>1</feature>
         <threshold>10.357875</threshold>
         <split pos="left">
            <feature>1</feature>
            <threshold>7.010513</threshold>
            <split pos="left">
               <feature>1</feature>
               <threshold>0.0</threshold>
               <split pos="left">
```

```
                <output>-1.131177544593811</output>
             </split>
             <split pos="right">
                <output>-1.131177544593811</output>
             </split>
          </split>
          <split pos="right">
             <output>-1.131177544593811</output>
          </split>
       </split>
       <split pos="right">
          <output>1.131177544593811</output>
       </split>
    </split>
 </tree>
 <tree id="9" weight="0.1">
    <split>
       <feature>2</feature>
       <threshold>10.573917</threshold>
       <split pos="left">
          <output>1.1046180725097656</output>
       </split>
       <split pos="right">
          <feature>1</feature>
          <threshold>7.010513</threshold>
          <split pos="left">
             <feature>1</feature>
             <threshold>0.0</threshold>
             <split pos="left">
                <output>-1.1046180725097656</output>
             </split>
             <split pos="right">
                <output>-1.1046180725097656</output>
             </split>
          </split>
          <split pos="right">
             <output>-1.1046180725097656</output>
          </split>
       </split>
    </split>
 </tree>
 <tree id="10" weight="0.1">
    <split>
       <feature>1</feature>
```

```
         <threshold>10.357875</threshold>
         <split pos="left">
            <feature>1</feature>
            <threshold>7.010513</threshold>
            <split pos="left">
               <feature>1</feature>
               <threshold>0.0</threshold>
               <split pos="left">
                  <output>-1.0838804244995117</output>
               </split>
               <split pos="right">
                  <output>-1.0838804244995117</output>
               </split>
            </split>
            <split pos="right">
               <output>-1.0838804244995117</output>
            </split>
         </split>
         <split pos="right">
            <output>1.0838804244995117</output>
         </split>
      </split>
   </tree>
</ensemble>
"""
    }
  }
}
```

Untuk melihat model, kirim permintaan berikut:

```
GET _ltr/_model/my_ranklib_model
```

Langkah 8: Lakukan pencarian dengan pembelajaran untuk peringkat

Setelah men-deploy model, Anda siap untuk mencari.

Ajukan kueri `sltr` dengan fitur yang Anda gunakan dan nama model yang ingin Anda jalankan:

```
POST tmdb/_search
{
  "_source": {
    "includes": ["title", "overview"]
```

```
  },
  "query": {
    "multi_match": {
      "query": "rambo",
      "fields": ["title", "overview"]
    }
  },
  "rescore": {
    "query": {
      "rescore_query": {
        "sltr": {
          "params": {
            "keywords": "rambo"
          },
          "model": "my_ranklib_model"
        }
      }
    }
  }
}
```

Dengan Learning to Rank, Anda melihat “Rambo” sebagai hasil pertama karena kita telah menetapkan nilai tertinggi dalam daftar penilaian:

```
{
  "took" : 12,
  "timed_out" : false,
  "_shards" : {
    "total" : 1,
    "successful" : 1,
    "skipped" : 0,
    "failed" : 0
  },
  "hits" : {
    "total" : {
      "value" : 7,
      "relation" : "eq"
    },
    "max_score" : 13.096414,
    "hits" : [
      {
        "_index" : "tmdb",
        "_type" : "movie",
```

```
        "_id" : "7555",
        "_score" : 13.096414,
        "_source" : {
          "overview" : "When governments fail to act on behalf of captive missionaries,
ex-Green Beret John James Rambo sets aside his peaceful existence along the Salween
River in a war-torn region of Thailand to take action.  Although he's still haunted
by violent memories of his time as a U.S. soldier during the Vietnam War, Rambo can
hardly turn his back on the aid workers who so desperately need his help.",
          "title" : "Rambo"
        }
      },
      {
        "_index" : "tmdb",
        "_type" : "movie",
        "_id" : "1370",
        "_score" : 11.17245,
        "_source" : {
          "overview" : "Combat has taken its toll on Rambo, but he's finally begun to
find inner peace in a monastery. When Rambo's friend and mentor Col. Trautman asks for
his help on a top secret mission to Afghanistan, Rambo declines but must reconsider
when Trautman is captured.",
          "title" : "Rambo III"
        }
      },
      {
        "_index" : "tmdb",
        "_type" : "movie",
        "_id" : "1368",
        "_score" : 10.442155,
        "_source" : {
          "overview" : "When former Green Beret John Rambo is harassed by local law
enforcement and arrested for vagrancy, the Vietnam vet snaps, runs for the hills and
rat-a-tat-tats his way into the action-movie hall of fame. Hounded by a relentless
sheriff, Rambo employs heavy-handed guerilla tactics to shake the cops off his tail.",
          "title" : "First Blood"
        }
      },
      {
        "_index" : "tmdb",
        "_type" : "movie",
        "_id" : "1369",
        "_score" : 10.442155,
        "_source" : {
```

```
        "overview" : "Col. Troutman recruits ex-Green Beret John Rambo for a highly
secret and dangerous mission. Teamed with Co Bao, Rambo goes deep into Vietnam to
rescue POWs. Deserted by his own team, he's left in a hostile jungle to fight for his
life, avenge the death of a woman and bring corrupt officials to justice.",
        "title" : "Rambo: First Blood Part II"
      }
    },
    {
      "_index" : "tmdb",
      "_type" : "movie",
      "_id" : "31362",
      "_score" : 7.424202,
      "_source" : {
        "overview" : "It is 1985, and a small, tranquil Florida town is being rocked
by a wave of vicious serial murders and bank robberies. Particularly sickening to the
authorities is the gratuitous use of violence by two “Rambo” like killers who dress
themselves in military garb. Based on actual events taken from FBI files, the movie
depicts the Bureau’s efforts to track down these renegades.",
        "title" : "In the Line of Duty: The F.B.I. Murders"
      }
    },
    {
      "_index" : "tmdb",
      "_type" : "movie",
      "_id" : "13258",
      "_score" : 6.43182,
      "_source" : {
        "overview" : """Will Proudfoot (Bill Milner) is looking for an escape from
his family's stifling home life when he encounters Lee Carter (Will Poulter), the
school bully. Armed with a video camera and a copy of "Rambo: First Blood", Lee plans
to make cinematic history by filming his own action-packed video epic. Together, these
two newfound friends-turned-budding-filmmakers quickly discover that their imaginative
– and sometimes mishap-filled – cinematic adventure has begun to take on a life of its
own!""",
        "title" : "Son of Rambow"
      }
    },
    {
      "_index" : "tmdb",
      "_type" : "movie",
      "_id" : "61410",
      "_score" : 3.9719706,
      "_source" : {
```

```
          "overview" : "It's South Africa 1990. Two major events are about to happen:
 The release of Nelson Mandela and, more importantly, it's Spud Milton's first year
 at an elite boys only private boarding school. John Milton is a boy from an ordinary
 background who wins a scholarship to a private school in Kwazulu-Natal, South Africa.
 Surrounded by boys with nicknames like Gecko, Rambo, Rain Man and Mad Dog, Spud has
 his hands full trying to adapt to his new home. Along the way Spud takes his first
 tentative steps along the path to manhood. (The path it seems could be a rather long
 road). Spud is an only child. He is cursed with parents from well beyond the lunatic
 fringe and a senile granny. His dad is a fervent anti-communist who is paranoid that
 the family domestic worker is running a shebeen from her room at the back of the
 family home. His mom is a free spirit and a teenager's worst nightmare, whether it's
 shopping for Spud's underwear in the local supermarket",
          "title" : "Spud"
        }
      }
    ]
  }
}
```

Jika Anda mencari tanpa menggunakan plugin Learning to Rank, OpenSearch mengembalikan hasil yang berbeda:

```
POST tmdb/_search
{
  "_source": {
    "includes": ["title", "overview"]
  },
  "query": {
    "multi_match": {
      "query": "Rambo",
      "fields": ["title", "overview"]
    }
  }
}
```

```
{
  "took" : 5,
  "timed_out" : false,
  "_shards" : {
    "total" : 1,
    "successful" : 1,
    "skipped" : 0,
    "failed" : 0
```

```
 },
 "hits" : {
   "total" : {
     "value" : 5,
     "relation" : "eq"
   },
   "max_score" : 11.262714,
   "hits" : [
     {
       "_index" : "tmdb",
       "_type" : "movie",
       "_id" : "1370",
       "_score" : 11.262714,
       "_source" : {
         "overview" : "Combat has taken its toll on Rambo, but he's finally begun to
find inner peace in a monastery. When Rambo's friend and mentor Col. Trautman asks for
his help on a top secret mission to Afghanistan, Rambo declines but must reconsider
when Trautman is captured.",
         "title" : "Rambo III"
       }
     },
     {
       "_index" : "tmdb",
       "_type" : "movie",
       "_id" : "7555",
       "_score" : 11.2569065,
       "_source" : {
         "overview" : "When governments fail to act on behalf of captive missionaries,
ex-Green Beret John James Rambo sets aside his peaceful existence along the Salween
River in a war-torn region of Thailand to take action.  Although he's still haunted
by violent memories of his time as a U.S. soldier during the Vietnam War, Rambo can
hardly turn his back on the aid workers who so desperately need his help.",
         "title" : "Rambo"
       }
     },
     {
       "_index" : "tmdb",
       "_type" : "movie",
       "_id" : "1368",
       "_score" : 10.558305,
       "_source" : {
         "overview" : "When former Green Beret John Rambo is harassed by local law
enforcement and arrested for vagrancy, the Vietnam vet snaps, runs for the hills and
```

```
 rat-a-tat-tats his way into the action-movie hall of fame. Hounded by a relentless
 sheriff, Rambo employs heavy-handed guerilla tactics to shake the cops off his tail.",
          "title" : "First Blood"
        }
      },
      {
        "_index" : "tmdb",
        "_type" : "movie",
        "_id" : "1369",
        "_score" : 10.558305,
        "_source" : {
          "overview" : "Col. Troutman recruits ex-Green Beret John Rambo for a highly
 secret and dangerous mission. Teamed with Co Bao, Rambo goes deep into Vietnam to
 rescue POWs. Deserted by his own team, he's left in a hostile jungle to fight for his
 life, avenge the death of a woman and bring corrupt officials to justice.",
          "title" : "Rambo: First Blood Part II"
        }
      },
      {
        "_index" : "tmdb",
        "_type" : "movie",
        "_id" : "13258",
        "_score" : 6.4600153,
        "_source" : {
          "overview" : """Will Proudfoot (Bill Milner) is looking for an escape from
 his family's stifling home life when he encounters Lee Carter (Will Poulter), the
 school bully. Armed with a video camera and a copy of "Rambo: First Blood", Lee plans
 to make cinematic history by filming his own action-packed video epic. Together, these
 two newfound friends-turned-budding-filmmakers quickly discover that their imaginative
 – and sometimes mishap-filled – cinematic adventure has begun to take on a life of its
 own!""",
          "title" : "Son of Rambow"
        }
      }
    ]
  }
}
```

Berdasarkan penilaian Anda terhadap performa model tersebut, sesuaikan daftar penilaian dan fiturnya. Kemudian, ulangi langkah 2-8 untuk meningkatkan hasil peringkat dari waktu ke waktu.

API Learning to Rank

Gunakan operasi Learning to Rank untuk bekerja dengan set dan model fitur secara terprogram.

Buat penyimpanan

Membuat indeks `.ltrstore` tersembunyi yang menyimpan informasi metadata seperti set dan model fitur.

```
PUT _ltr
```

Hapus penyimpanan

Menghapus indeks `.ltrstore` tersembunyi dan me-reset plugin.

```
DELETE _ltr
```

Buat set fitur

Membuat satu set fitur.

```
POST _ltr/_featureset/<name_of_features>
```

Hapus set fitur

Menghapus satu set fitur.

```
DELETE _ltr/_featureset/<name_of_feature_set>
```

Dapatkan set fitur

Mengambil satu set fitur.

```
GET _ltr/_featureset/<name_of_feature_set>
```

Buat Model

Membuat model.

```
POST _ltr/_featureset/<name_of_feature_set>/_createmodel
```

Hapus model

Menghapus model.

```
DELETE _ltr/_model/<name_of_model>
```

Dapatkan model

Mengambil model.

```
GET _ltr/_model/<name_of_model>
```

Dapatkan statistik

Memberikan informasi tentang perilaku plugin.

```
GET _ltr/_stats
```

Anda juga dapat menggunakan filter untuk mengambil satu stat:

```
GET _ltr/_stats/<stat>
```

Futthermore, Anda dapat membatasi informasi ke satu node di cluster:

```
GET _ltr/_stats/<stat>/nodes/<nodeId>

{
  "_nodes" : {
    "total" : 1,
    "successful" : 1,
    "failed" : 0
  },
  "cluster_name" : "873043598401:ltr-77",
  "stores" : {
    ".ltrstore" : {
      "model_count" : 1,
      "featureset_count" : 1,
      "feature_count" : 2,
      "status" : "green"
```

```
    }
  },
  "status" : "green",
  "nodes" : {
    "DjelK-_ZSfyzstO5dhGGQA" : {
      "cache" : {
        "feature" : {
          "eviction_count" : 0,
          "miss_count" : 0,
          "entry_count" : 0,
          "memory_usage_in_bytes" : 0,
          "hit_count" : 0
        },
        "featureset" : {
          "eviction_count" : 2,
          "miss_count" : 2,
          "entry_count" : 0,
          "memory_usage_in_bytes" : 0,
          "hit_count" : 0
        },
        "model" : {
          "eviction_count" : 2,
          "miss_count" : 3,
          "entry_count" : 1,
          "memory_usage_in_bytes" : 3204,
          "hit_count" : 1
        }
      },
      "request_total_count" : 6,
      "request_error_count" : 0
    }
  }
}
```

Statistik disediakan pada dua tingkat, simpul, dan klaster, seperti yang ditentukan dalam tabel berikut:

Statistik tingkat-simpul

Nama kolom	Deskripsi
request_total_count	Jumlah total permintaan peringkat.
request_error_count	Jumlah total permintaan gagal.

Nama kolom	Deskripsi
Cache	Statistik di semua cache (fitur, setfitur, model). Sebuah temuan cache terjadi ketika pengguna mengajukan kueri plugin dan model sudah dimuat ke dalam memori.
cache.eviction_count	Jumlah pengosongan cache.
cache.hit_count	Jumlah temuan cache.
cache.miss_count	Jumlah cache meleset. Sebuah cache miss terjadi ketika pengguna mengajukan kueri plugin dan model belum dimuat ke dalam memori.
cache.entry_count	Jumlah entri dalam cache.
cache.memory_usage_in_bytes	Total memori yang digunakan dalam byte.
cache.cache_capacity_reached	Menunjukkan bahwa batas cache tercapai.

Statistik tingkat-klaster

Nama kolom	Deskripsi
penyimpanan	Menunjukkan di mana set fitur dan metadata model disimpan. (Default adalah “.ltrstore”. Jika tidak, itu diawali dengan “.ltrstore_”, dengan nama yang disediakan pengguna).
stores.status	Status indeks.
stores.feature_sets	Jumlah set fitur.
stores.features_count	Jumlah fitur.
stores.model_count	Jumlah model.

Nama kolom	Deskripsi
status	Status plugin berdasarkan status indeks tempat penyimpanan fitur (merah, kuning, atau hijau) dan keadaan pemutus sirkuit (terbuka atau tertutup).
cache.cache_capacity_reached	Menunjukkan bahwa batasan cache tercapai.

Dapatkan statistik cache

Mengembalikan statistik tentang penggunaan cache dan memori.

```
GET _ltr/_cachestats

{
    "_nodes": {
        "total": 2,
        "successful": 2,
        "failed": 0
    },
    "cluster_name": "opensearch-cluster",
    "all": {
        "total": {
            "ram": 612,
            "count": 1
        },
        "features": {
            "ram": 0,
            "count": 0
        },
        "featuresets": {
            "ram": 612,
            "count": 1
        },
        "models": {
            "ram": 0,
            "count": 0
        }
    },
    "stores": {
```

```
        ".ltrstore": {
            "total": {
                "ram": 612,
                "count": 1
            },
            "features": {
                "ram": 0,
                "count": 0
            },
            "featuresets": {
                "ram": 612,
                "count": 1
            },
            "models": {
                "ram": 0,
                "count": 0
            }
        }
    },
    "nodes": {
        "ejF6uutERF20wOFNOXB61A": {
            "name": "opensearch1",
            "hostname": "172.18.0.4",
            "stats": {
                "total": {
                    "ram": 612,
                    "count": 1
                },
                "features": {
                    "ram": 0,
                    "count": 0
                },
                "featuresets": {
                    "ram": 612,
                    "count": 1
                },
                "models": {
                    "ram": 0,
                    "count": 0
                }
            }
        },
        "Z2RZNWRLSveVcz2c6lHf5A": {
            "name": "opensearch2",
```

```
            "hostname": "172.18.0.2",
            "stats": {
                ...
            }
        }
    }
}
```

Hapus cache

Membersihkan cache plugin. Gunakan ini untuk menyegarkan model.

```
POST _ltr/_clearcache
```

Pencarian asinkron di Amazon Service OpenSearch

Dengan penelusuran asinkron untuk Amazon OpenSearch Service, Anda dapat mengirimkan kueri penelusuran yang dijalankan di latar belakang, memantau kemajuan permintaan, dan mengambil hasil di tahap selanjutnya. Anda dapat mengambil sebagian hasil karena hasil tersebut menjadi tersedia sebelum pencarian selesai. Setelah pencarian selesai, simpan hasil untuk pengambilan dan analisis nanti.

Pencarian asinkron membutuhkan OpenSearch 1.0 atau yang lebih baru, atau Elasticsearch 7.10 atau yang lebih baru. Dokumentasi lengkap untuk penelusuran asinkron, termasuk langkah-langkah rinci dan deskripsi API, tersedia dalam dokumentasi. OpenSearch

Contoh panggilan pencarian

Untuk melakukan pencarian asinkron, kirim permintaan HTTP ke `_plugins/_asynchronous_search` menggunakan format berikut:

```
POST opensearch-domain/_plugins/_asynchronous_search
```

> **Note**
>
> Jika Anda menggunakan Elasticsearch 7.10 alih-alih OpenSearch versi, ganti `_plugins` dengan semua permintaan penelusuran `_opendistro` asinkron.

Anda dapat menentukan opsi pencarian asinkron berikut:

Opsi	Deskripsi	Nilai default	Diperlukan
`wait_for_completion_timeout`	Menentukan jumlah waktu yang Anda rencanakan untuk menunggu hasil. Anda dapat melihat hasil apa pun yang Anda dapatkan dalam waktu ini seperti dalam pencarian normal. Anda dapat melakukan poling hasil yang tersisa berdasarkan ID. Nilai maksimum yang diizinkan adalah 300 detik.	1 detik	Tidak
`keep_on_completion`	Menentukan apakah Anda ingin menyimpan hasil dalam klaster setelah pencarian selesai. Anda dapat memeriksa hasil yang disimpan di lain waktu.	`false`	Tidak
`keep_alive`	Menentukan jumlah waktu hasilnya disimpan dalam klaster. Misalnya, `2d` berarti bahwa hasilnya disimpan di dalam klaster selama 48 jam. Hasil pencarian yang disimpan akan dihapus setelah periode ini atau jika pencarian dibatalkan. Perhatikan bahwa ini termasuk waktu aktif kueri. Jika permintaan overruns saat ini, proses membatalkan kueri ini secara otomatis.	`12 jam`	Tidak

Permintaan sampel

```
POST _plugins/_asynchronous_search/?
pretty&size=10&wait_for_completion_timeout=1ms&keep_on_completion=true&request_cache=false
{
  "aggs": {
    "city": {
      "terms": {
        "field": "city",
        "size": 10
      }
```

```
    }
  }
}
```

> **Note**
>
> Semua parameter permintaan yang berlaku untuk kueri `_search` standar yang didukung. Jika Anda menggunakan Elasticsearch 7.10 alih-alih OpenSearch versi, ganti dengan. `_plugins` `_opendistro`

Izin pencarian asinkron

Pencarian asinkron mendukung kontrol akses berbutir halus. Untuk detail tentang pencampuran dan pencocokan izin agar sesuai dengan kasus penggunaan Anda, lihat Pencarian asinkron.

Untuk domain dengan kontrol akses berbutir halus diaktifkan, Anda memerlukan izin minimum berikut untuk peran:

```
# Allows users to use all asynchronous search functionality
asynchronous_search_full_access:
  reserved: true
  cluster_permissions:
    - 'cluster:admin/opensearch/asynchronous-search/*'
  index_permissions:
    - index_patterns:
        - '*'
      allowed_actions:
        - 'indices:data/read/search*'

# Allows users to read stored asynchronous search results
asynchronous_search_read_access:
  reserved: true
  cluster_permissions:
    - 'cluster:admin/opensearch/asynchronous-search/get'
```

Untuk domain dengan kontrol akses berbutir halus dinonaktifkan, gunakan akses IAM dan kunci rahasia untuk menandatangani semua permintaan. Anda dapat mengakses hasil dengan ID pencarian asinkron.

Pengaturan pencarian asinkron

OpenSearch memungkinkan Anda mengubah semua setelan pencarian asinkron yang tersedia menggunakan API. `_cluster/settings` Di OpenSearch Layanan, Anda hanya dapat mengubah pengaturan berikut:

- `plugins.asynchronous_search.node_concurrent_running_searches`
- `plugins.asynchronous_search.persist_search_failures`

Pencarian lintas klaster

Anda dapat melakukan pencarian asinkron di klaster dengan keterbatasan kecil berikut:

- Anda dapat menjalankan pencarian asinkron hanya pada domain sumber.
- Anda tidak dapat meminimalkan perjalanan putaran jaringan sebagai bagian dari kueri pencarian lintas-klaster.

Jika Anda mengatur koneksi antara `domain-a -> domain-b` dengan alias `cluster_b` dan `domain-a -> domain-c` dengan alias koneksi `cluster_c`, pencarian asinkron `domain-a`, `domain-b`, dan `domain-c` sebagai berikut:

```
POST https://src-domain.us-east-1.es.amazonaws.com/
local_index,cluster_b:b_index,cluster_c:c_index/_plugins/_asynchronous_search/?
pretty&size=10&wait_for_completion_timeout=500ms&keep_on_completion=true&request_cache=false
{
  "size": 0,
  "_source": {
    "excludes": []
  },
  "aggs": {
    "2": {
      "terms": {
        "field": "clientip",
        "size": 50,
        "order": {
          "_count": "desc"
        }
      }
    }
  },
```

```
  "stored_fields": [
    "*"
  ],
  "script_fields": {},
  "docvalue_fields": [
    "@timestamp"
  ],
  "query": {
    "bool": {
      "must": [
        {
          "query_string": {
            "query": "status:404",
            "analyze_wildcard": true,
            "default_field": "*"
          }
        },
        {
          "range": {
            "@timestamp": {
              "gte": 1483747200000,
              "lte": 1488326400000,
              "format": "epoch_millis"
            }
          }
        }
      ],
      "filter": [],
      "should": [],
      "must_not": []
    }
  }
}
```

Respon

```
{
  "id" :
 "Fm9pYzJyVG91U19xb0hIQUJnMHJfRFEAAAAAAAknghQ1OWVBczNZQjVEa2dMYTBXaTdEagAAAAAAAAAB",
  "state" : "RUNNING",
  "start_time_in_millis" : 1609329314796,
  "expiration_time_in_millis" : 1609761314796
}
```

Untuk informasi selengkapnya, lihat the section called “Pencarian lintas klaster”.

UltraWarm

Pencarian asinkron dengan UltraWarm indeks terus berfungsi. Untuk informasi selengkapnya, lihat the section called “UltraWarm penyimpanan”.

> **Note**
>
> Anda dapat memantau statistik pencarian asinkron di. CloudWatch Untuk daftar lengkap metrik, lihat the section called “Metrik pencarian asinkron”.

Titik waktu di Amazon OpenSearch Service

Fitur point in time (PIT) adalah jenis pencarian yang memungkinkan Anda menjalankan kueri berbeda terhadap kumpulan data yang diperbaiki tepat waktu. Biasanya, ketika Anda menjalankan kueri yang sama pada indeks yang sama pada titik waktu yang berbeda, Anda menerima hasil yang berbeda karena dokumen terus diindeks, diperbarui, dan dihapus. Dengan PIT, Anda dapat query terhadap keadaan konstan dataset Anda.

Penggunaan utama fitur PIT adalah memasangkannya dengan`search_after` fungsionalitas. Ini adalah metode pagination yang disukai OpenSearch, terutama untuk pagination dalam, karena beroperasi pada dataset yang dibekukan dalam waktu, itu tidak terikat pada query, dan mendukung pagination konsisten maju dan mundur. Anda dapat menggunakan PIT dengan OpenSearch Service versi 2.5 dan yang lebih baru.

Untuk informasi lebih lanjut tentang PIT, lihat Point in Time dalam OpenSearch dokumentasi.

Pertimbangan-pertimbangan

Pertimbangkan sebagai berikut ketika Anda mengkonfigurasi pencarian PIT Anda:

- Jika Anda meningkatkan dari domain 2.3 dan memerlukan kontrol akses butiran halus pada tindakan PIT, Anda perlu menambahkan tindakan dan peran tersebut secara manual.
- Tidak ada ketahanan untuk PIT. Node reboot, penghentian node, penyebaran biru/hijau, dan proses ES restart menyebabkan semua data PIT hilang.

- Jika pecahan pindah selama penyebaran biru/hijau, hanya segmen data langsung yang ditransfer ke node baru. Segmen pecahan yang dipegang oleh PIT (baik secara eksklusif maupun yang dibagikan dengan data hidup) tetap berada di simpul lama.
- Pencarian PIT saat ini tidak berfungsi dengan pencarian asinkron.

Buat PIT

Untuk membuat PIT, kirim permintaan HTTP untuk`_search/point_in_time` menggunakan format berikut:

```
POST opensearch-domain/my-index/_search/point_in_time?keep_alive=time
```

Anda dapat menentukan opsi PIT berikut:

Opsi	Deskripsi	Nilai default	Diperlukan
`keep_alive`	Jumlah waktu untuk menjaga PIT. Setiap kali Anda mengakses PIT dengan permintaan pencarian, masa pakai PIT diperpanjang dengan jumlah waktu yang sama dengan`keep_alive` parameter. Parameter kueri ini diperlukan saat Anda membuat PIT, tetapi opsional dalam permintaan pencarian.		Ya
`preference`	Sebuah string yang menentukan node atau pecahan yang digunakan untuk melakukan pencarian.	Acak	Tidak
`routing`	String yang menentukan untuk merutekan permintaan pencarian ke pecahan tertentu.	Dokumenny`a_id`	Tidak
`expand_wildcards`	String yang menentukan jenis indeks yang dapat mencocokkan pola wildcard. Mendukung nilai yang dipisahkan koma. Nilai yang benar adalah sebagai berikut: • `all`: Cocokkan indeks atau aliran data apa pun, termasuk yang tersembunyi.	`open`	Tidak

Opsi	Deskripsi	Nilai default	Diperlukan
	• `open`: Cocokkan indeks terbuka, tidak tersembunyi, atau aliran data yang tidak tersembunyi. • `closed`: Cocokkan indeks tertutup, tidak tersembunyi, atau aliran data yang tidak tersembunyi. • `hidden`: Cocokkan indeks tersembunyi atau aliran data. Harus dikombinasikan dengan terbuka, tertutup atau terbuka dan tertutup. • `none`: Tidak ada pola wildcard yang diterima.		
`allow_partial_pit_creation`	Sebuah boolean yang menentukan apakah akan membuat PIT dengan kegagalan sebagian.	`true`	Tidak

Respon sampel

```
{
    "pit_id":
 "o463QQEPbXktaW5kZXgtMDAwMDAxFnNOWU43ckt3U3IyaFVpbGE1UWEtMncAFjFyeXBsRGJmVFM2RTB6eVg1aVVqQncAA
    "_shards": {
        "total": 1,
        "successful": 1,
        "skipped": 0,
        "failed": 0
    },
    "creation_time": 1658146050064
}
```

Ketika Anda membuat PIT, Anda menerima ID PIT di respon. Ini adalah ID yang Anda gunakan untuk melakukan pencarian dengan PIT.

Izin titik waktu

PIT mendukung kontrol akses detail. Jika Anda meningkatkan ke domain 2,5 dan memerlukan kontrol akses butiran halus, Anda perlu membuat peran secara manual dengan izin berikut:

```
# Allows users to use all point in time search search functionality
point_in_time_full_access:
  reserved: true
  index_permissions:
    - index_patterns:
        - '*'
      allowed_actions:
        - "indices:data/read/point_in_time/create"
        - "indices:data/read/point_in_time/delete"
        - "indices:data/read/point_in_time/readall"
        - "indices:data/read/search"
        - "indices:monitor/point_in_time/segments"


# Allows users to use point in time search search functionality for specific index
# All type operations like list all PITs, delete all PITs are not supported in this
 case

point_in_time_index_access:
  reserved: true
  index_permissions:
    - index_patterns:
        - 'my-index-1'
      allowed_actions:
        - "indices:data/read/point_in_time/create"
        - "indices:data/read/point_in_time/delete"
        - "indices:data/read/search"
        - "indices:monitor/point_in_time/segments"
```

Untuk domain dengan versi 2.5 ke atas, Anda dapat menggunakan`point_in_time_full_access` peran bawaan. Untuk informasi selengkapnya, lihat Model keamanan di OpenSearch dokumentasi.

Pengaturan PIT

OpenSearch memungkinkan Anda mengubah semua pengaturan PIT yang tersedia menggunakan`_cluster/settings` API. Di OpenSearch Layanan, saat ini Anda tidak dapat mengubah pengaturan.

Pencarian lintas klaster

Anda dapat membuat PITs, mencari dengan PIT ID, daftar PITs, dan menghapus PITs di seluruh cluster dengan batasan kecil berikut:

- Anda dapat mencantumkan semua dan menghapus semua lubang hanya pada domain sumber.
- Anda tidak dapat meminimalkan perjalanan putaran jaringan sebagai bagian dari kueri pencarian lintas-klaster.

Untuk informasi selengkapnya, lihat the section called “Pencarian lintas klaster”.

UltraWarm

Pencarian PIT dengan UltraWarm indeks terus bekerja. Untuk informasi selengkapnya, lihat the section called “UltraWarm penyimpanan”.

> **Note**
>
> Anda dapat memantau statistik pencarian PIT di CloudWatch. Untuk daftar lengkap metrik, lihat the section called “Metrik titik dalam waktu”.

Pencarian semantik di Layanan Amazon OpenSearch

Dimulai dengan OpenSearch Layanan versi 2.9, Anda dapat menggunakan pencarian semantik untuk membantu Anda memahami kueri penelusuran dan meningkatkan relevansi penelusuran. Anda dapat menggunakan pencarian semantik dengan salah satu dari dua cara — dengan pencarian saraf dan dengan k-NN.

Dengan OpenSearch Service, Anda dapat mengatur konektor AI untuk Layanan AWS dan layanan eksternal. Menggunakan konsol, Anda juga dapat membuat model ML dengan AWS CloudFormation template. Lihat informasi yang lebih lengkap di the section called “CloudFormation integrasi template”.

Menggunakan OpenSearch Dasbor dengan Layanan Amazon OpenSearch

OpenSearch Dasbor adalah alat visualisasi sumber terbuka yang dirancang untuk digunakan. OpenSearch Amazon OpenSearch Service menyediakan instalasi OpenSearch Dasbor dengan setiap domain OpenSearch Layanan.

Anda dapat menemukan tautan ke OpenSearch Dasbor di dasbor domain Anda di konsol OpenSearch Layanan. Untuk domain yang berjalan OpenSearch, URL adalah*domain-endpoint*/_dashboards/. Untuk domain yang menjalankan Elasticsearch lama, URL-nya adalah. *domain-endpoint*/_plugin/kibana

Kueri yang menggunakan instalasi OpenSearch Dasbor default ini memiliki batas waktu 300 detik.

Bagian berikut membahas beberapa kasus penggunaan umum untuk OpenSearch Dasbor:

- the section called “Mengontrol akses ke OpenSearch Dasbor”
- the section called “Mengkonfigurasi OpenSearch Dasbor untuk menggunakan server peta WMS”
- the section called “Menghubungkan server Dasbor lokal ke Layanan OpenSearch ”

Mengontrol akses ke OpenSearch Dasbor

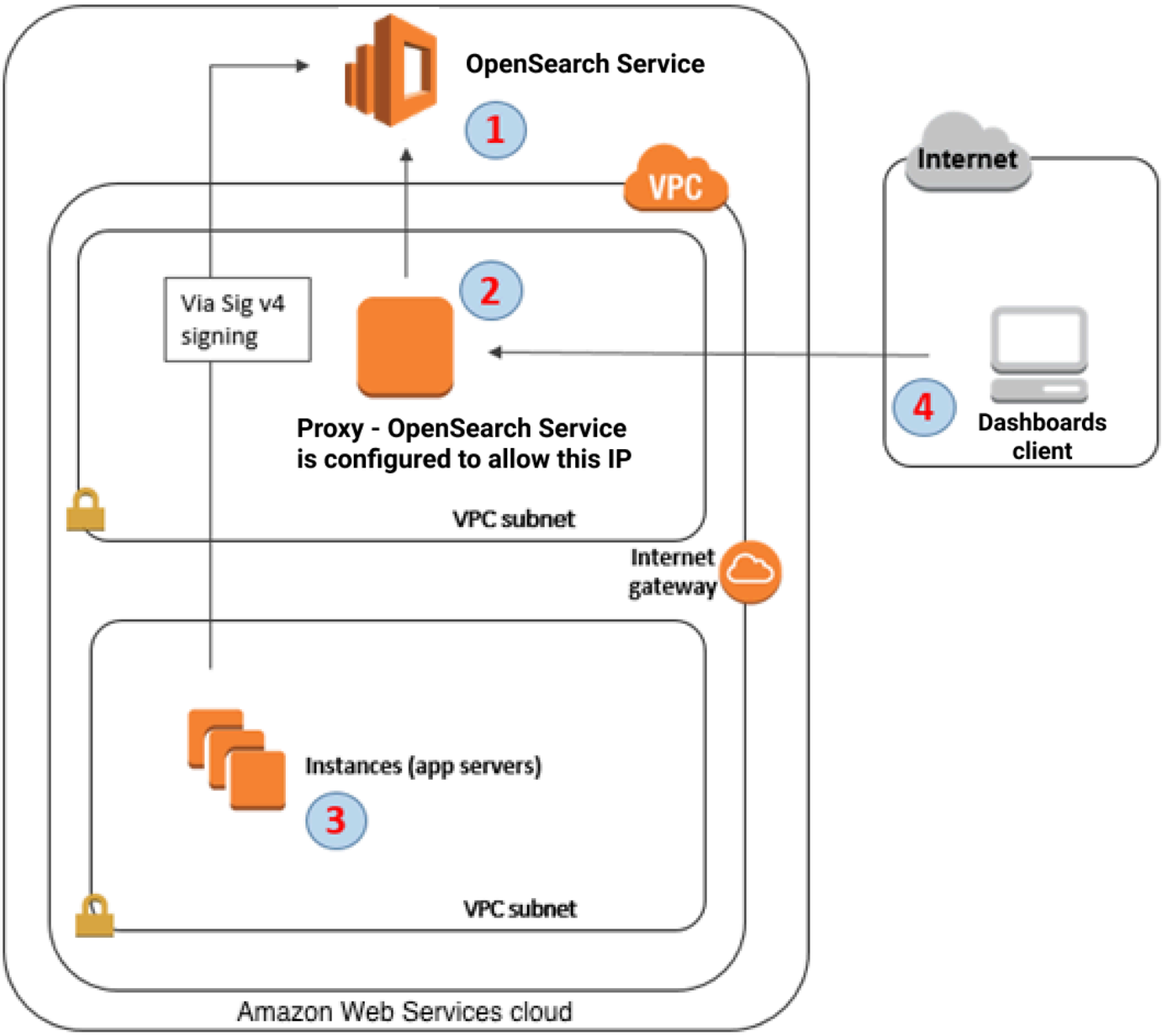
Dasbor tidak mendukung pengguna dan peran IAM secara native, tetapi OpenSearch Layanan menawarkan beberapa solusi untuk mengontrol akses ke Dasbor:

- Aktifkan otentikasi SAMP untuk Dasbor.
- Gunakan Kontrol akses detail dengan autentikasi dasar HTTP.
- Konfigurasikan otentikasi Cognito untuk Dasbor.
- Untuk domain akses publik, konfigurasikan kebijakan akses berbasis IP yang menggunakan atau tidak menggunakan server proxy.
- Untuk domain akses VPC, gunakan kebijakan akses terbuka yang menggunakan atau tidak menggunakan server proxy, dan grup keamanan untuk mengontrol akses. Untuk mempelajari selengkapnya, lihat the section called “Tentang kebijakan akses pada domain VPC”.

Menggunakan proxy untuk mengakses OpenSearch Layanan dari OpenSearch Dasbor

> **Note**
>
> Proses ini hanya berlaku jika domain Anda menggunakan akses publik dan Anda tidak ingin menggunakan otentikasi Cognito. Lihat the section called “Mengontrol akses ke OpenSearch Dasbor”.

Karena Dasbor adalah JavaScript aplikasi, permintaan berasal dari alamat IP pengguna. Kontrol akses berbasis IP mungkin tidak praktis karena banyaknya alamat IP yang perlu Anda izinkan agar setiap pengguna memiliki akses ke Dasbor. Salah satu solusinya adalah menempatkan server proxy antara OpenSearch Dasbor dan Layanan. OpenSearch Kemudian Anda dapat menambahkan kebijakan akses berbasis IP yang memungkinkan permintaan dari hanya satu alamat IP, proksi. Diagram berikut menunjukkan konfigurasi ini.



1. Ini adalah domain OpenSearch Layanan Anda. IAM menyediakan akses resmi ke domain ini. Kebijakan akses tambahan berbasis IP menyediakan akses ke server proksi.
2. Ini adalah server proksi, berjalan di instans Amazon EC2.
3. Aplikasi lain dapat menggunakan proses penandatanganan Signature Version 4 untuk mengirim permintaan yang diautentikasi ke OpenSearch Layanan.
4. OpenSearch Klien dasbor terhubung ke domain OpenSearch Layanan Anda melalui proxy.

Untuk mengaktifkan konfigurasi semacam ini, Anda memerlukan kebijakan berbasis sumber daya yang menentukan peran dan alamat IP. Berikut adalah contoh kebijakan:

```
{
  "Version": "2012-10-17",
  "Statement": [{
      "Resource": "arn:aws:es:us-west-2:111111111111:domain/my-domain/*",
      "Principal": {
        "AWS": "arn:aws:iam::111111111111:role/allowedrole1"
      },
      "Action": [
        "es:ESHttpGet"
      ],
      "Effect": "Allow"
    },
    {
      "Effect": "Allow",
      "Principal": {
        "AWS": "*"
      },
      "Action": "es:*",
      "Condition": {
        "IpAddress": {
          "aws:SourceIp": [
            "123.456.789.123"
          ]
        }
      },
      "Resource": "arn:aws:es:us-west-2:111111111111:domain/my-domain/*"
    }
  ]
}
```

Kami merekomendasikan bahwa Anda mengonfigurasi contoh EC2 menjalankan server proksi dengan alamat IP Elastis. Dengan cara ini, Anda dapat mengganti instans bila diperlukan dan masih melampirkan alamat IP publik yang sama untuk itu. Untuk informasi selengkapnya, lihat AElastic IP Addresses di Panduan Pengguna Amazon EC2 untuk Instans Linux.

Jika Anda menggunakan server proxy dan otentikasi Cognito, Anda mungkin perlu menambahkan pengaturan untuk Dasbor dan Amazon Cognito untuk menghindari kesalahan. `redirect_mismatch` Lihat contoh `nginx.conf` berikut ini:

```
server {
    listen 443;
    server_name $host;
    rewrite ^/$ https://$host/_plugin/_dashboards redirect;

    ssl_certificate           /etc/nginx/cert.crt;
    ssl_certificate_key       /etc/nginx/cert.key;

    ssl on;
    ssl_session_cache  builtin:1000  shared:SSL:10m;
    ssl_protocols  TLSv1 TLSv1.1 TLSv1.2;
    ssl_ciphers HIGH:!aNULL:!eNULL:!EXPORT:!CAMELLIA:!DES:!MD5:!PSK:!RC4;
    ssl_prefer_server_ciphers on;

    location /_plugin/_dashboards {
        # Forward requests to Dashboards
        proxy_pass https://$dashboards_host/_plugin/_dashboards;

        # Handle redirects to Cognito
        proxy_redirect https://$cognito_host https://$host;

        # Update cookie domain and path
        proxy_cookie_domain $dashboards_host $host;
        proxy_cookie_path / /_plugin/_dashboards/;

        # Response buffer settings
        proxy_buffer_size 128k;
        proxy_buffers 4 256k;
        proxy_busy_buffers_size 256k;
    }

    location ~ \/(log|sign|fav|forgot|change|saml|oauth2) {
        # Forward requests to Cognito
        proxy_pass https://$cognito_host;

        # Handle redirects to Dashboards
        proxy_redirect https://$dashboards_host https://$host;

        # Update cookie domain
        proxy_cookie_domain $cognito_host $host;
    }
}
```

Mengkonfigurasi OpenSearch Dasbor untuk menggunakan server peta WMS

Instalasi default OpenSearch Dasbor untuk OpenSearch Layanan mencakup layanan peta, kecuali untuk domain di Wilayah India dan China. Layanan peta mendukung hingga 10 tingkat zoom.

Terlepas dari Wilayah Anda, Anda dapat mengonfigurasi Dasbor untuk menggunakan server Layanan Peta Web (WMS) yang berbeda untuk visualisasi peta koordinat. Visualisasi peta wilayah hanya mendukung layanan peta default.

Untuk mengkonfigurasi Dasbor untuk menggunakan server peta WMS:

1. Buka Dasbor.
2. Pilih Manajemen Stack.
3. Pilih Pengaturan Lanjutan.
4. Cari visualization:tileMap:WMSdefaults.
5. Ubah `enabled` ke `true` dan `url` ke URL server peta WMS yang valid:

```
{
  "enabled": true,
  "url": "wms-server-url",
  "options": {
    "format": "image/png",
    "transparent": true
  }
}
```

6. Pilih Simpan perubahan.

Untuk menerapkan nilai default baru ke visualisasi, Anda mungkin perlu memuat ulang Dasbor. Jika Anda telah menyimpan visualisasi, pilih Opsi setelah membuka visualisasi. Verifikasi bahwa Server peta WMS diaktifkan dan url WMS berisi server peta pilihan Anda, dan kemudian pilih Terapkan perubahan.

> **Note**
>
> Layanan peta sering kali memiliki biaya lisensi atau pembatasan. Anda bertanggung jawab atas semua pertimbangan tersebut pada setiap server peta yang Anda tentukan. Anda dapat menemukan layanan peta dari Survei Geologi AS yang berguna untuk pengujian.

Menghubungkan server Dasbor lokal ke Layanan OpenSearch

Jika Anda telah menginvestasikan waktu yang signifikan untuk mengonfigurasi instans OpenSearch Dasbor Anda sendiri, Anda dapat menggunakannya alih-alih (atau sebagai tambahan) instance Dasbor default yang OpenSearch disediakan Layanan. Prosedur berikut berfungsi untuk domain yang menggunakan kontrol akses berbutir halus dengan kebijakan akses terbuka.

Untuk menghubungkan server OpenSearch Dasbor lokal ke Layanan OpenSearch

1. Di domain OpenSearch Layanan Anda, buat pengguna dengan izin yang sesuai:
   a. Di Dasbor, buka Keamanan, Pengguna internal, dan pilih Buat pengguna internal.
   b. Berikan nama pengguna dan kata sandi dan pilih Buat.
   c. Masuk ke Peran dan pilih peran.
   d. Pilih Pengguna yang Dipetakan dan pilih Kelola pemetaan.
   e. Di Pengguna, tambahkan nama pengguna dan pilih Peta.
2. Unduh dan instal versi plugin OpenSearch keamanan yang sesuai pada instalasi OSS Dasbor yang dikelola sendiri.
3. Di server Dasbor lokal Anda, buka `config/opensearch_dashboards.yml` file dan tambahkan titik akhir OpenSearch Layanan Anda dengan nama pengguna dan kata sandi yang Anda buat sebelumnya:

   ```
   opensearch.hosts: ['https://domain-endpoint']
   opensearch.username: 'username'
   opensearch.password: 'password'
   ```

   Anda dapat menggunakan sampel file `opensearch_dashboards.yml` berikut:

   ```
   server.host: '0.0.0.0'
   ```

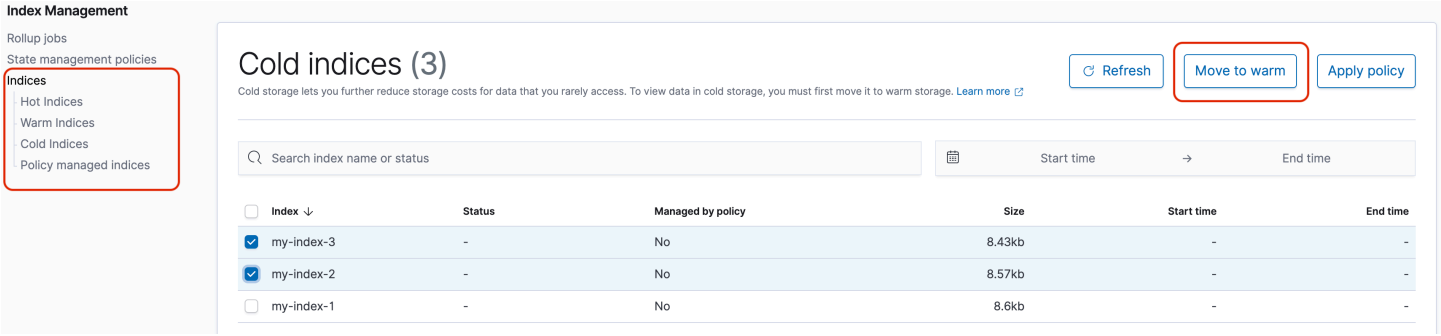
```
opensearch.hosts: ['https://domain-endpoint']

opensearch_dashboards.index: ".username"

opensearch.ssl.verificationMode: none # if not using HTTPS

opensearch_security.auth.type: basicauth
opensearch_security.auth.anonymous_auth_enabled: false
opensearch_security.cookie.secure: false # set to true when using HTTPS
opensearch_security.cookie.ttl: 3600000
opensearch_security.session.ttl: 3600000
opensearch_security.session.keepalive: false
opensearch_security.multitenancy.enabled: false
opensearch_security.readonly_mode.roles: ['opensearch_dashboards_read_only']
opensearch_security.auth.unauthenticated_routes: []
opensearch_security.basicauth.login.title: 'Please log in using your username and
 password'

opensearch.username: 'username'
opensearch.password: 'password'
opensearch.requestHeadersWhitelist:
[
authorization,
securitytenant,
security_tenant,
]
```

Untuk melihat indeks OpenSearch Layanan Anda, mulai server Dasbor lokal Anda, buka Alat Pengembang dan jalankan perintah berikut:

```
GET _cat/indices
```

Mengelola indeks di Dasbor OpenSearch

Instalasi OpenSearch Dasbor di domain OpenSearch Layanan Anda menyediakan UI yang berguna untuk mengelola indeks di berbagai tingkatan penyimpanan di domain Anda. Pilih Manajemen Indeks dari menu utama Dasbor untuk melihat semua indeks dalam penyimpanan panas, dan dingin UltraWarm, serta indeks yang dikelola oleh kebijakan Index State Management (ISM). Gunakan manajemen indeks untuk memindahkan indeks antara penyimpanan hangat dan dingin, dan untuk memantau migrasi antara tiga tingkatan.



Perhatikan bahwa Anda tidak akan melihat opsi indeks panas, hangat, dan dingin kecuali Anda mengaktifkan UltraWarm dan/atau penyimpanan dingin.

Fitur tambahan

Instalasi OpenSearch Dasbor default pada setiap domain OpenSearch Layanan memiliki beberapa fitur tambahan:

- Antarmuka pengguna untuk berbagai plugin OpenSearch
- Penyewa
- Laporan

  Gunakan menu Laporan untuk membuat laporan CSV sesuai permintaan dari halaman Discover dan laporan PDF atau PNG dasbor atau visualisasi. Laporan CSV memiliki batas 10.000 baris.
- Grafik Gantt
- Notebook

Mengelola indeks di Amazon Service OpenSearch

Setelah menambahkan data ke Amazon OpenSearch Service, Anda sering perlu mengindeks ulang data tersebut, bekerja dengan alias indeks, memindahkan indeks ke penyimpanan yang lebih hemat biaya, atau menghapusnya sama sekali. Bab ini mencakup UltraWarm penyimpanan, penyimpanan dingin, dan Manajemen Negara Indeks. Untuk informasi tentang API OpenSearch indeks, lihat OpenSearch dokumentasi.

Topik

UltraWarm penyimpanan untuk Amazon OpenSearch Service

UltraWarm menyediakan cara hemat biaya untuk menyimpan sejumlah besar data hanya-baca di Amazon Service. OpenSearch Simpul data standar menggunakan penyimpanan “panas”, yang berbentuk penyimpanan instans atau volume Amazon EBS yang dilampirkan pada setiap simpul. Penyimpanan panas memberikan performa tercepat untuk pengindeksan dan pencarian data baru.

Alih-alih penyimpanan terpasang, UltraWarm node menggunakan Amazon S3 dan solusi caching canggih untuk meningkatkan kinerja. Untuk indeks yang tidak Anda tulis secara aktif, kueri lebih jarang, dan tidak memerlukan kinerja yang sama, UltraWarm menawarkan biaya per GiB data yang jauh lebih rendah. Karena indeks hangat hanya-baca kecuali Anda mengembalikannya ke penyimpanan panas, paling cocok untuk data UltraWarm yang tidak dapat diubah, seperti log.

Dalam OpenSearch, indeks hangat berperilaku seperti indeks lainnya. Anda dapat melakukan kueri menggunakan API yang sama atau menggunakannya untuk membuat visualisasi di OpenSearch Dasbor.

Topik

Prasyarat

UltraWarm memiliki beberapa prasyarat penting:

- UltraWarm membutuhkan OpenSearch atau Elasticsearch 6.8 atau lebih tinggi.
- Untuk menggunakan penyimpanan hangat, domain harus memiliki simpul utama terdedikasi.
- Jika domain Anda menggunakan tipe instans T2 atau T3 untuk simpul data, Anda tidak dapat menggunakan penyimpanan hangat.
- Jika indeks Anda menggunakan codec kompresi Zstandard (`"index.codec": "zstd"`atau`"index.codec": "zstd_no_dict"`), Anda tidak dapat memindahkannya ke penyimpanan hangat.
- Jika indeks Anda menggunakan perkiraan k-nn (`"index.knn": true`), Anda tidak dapat memindahkannya ke penyimpanan hangat.
- Jika domain menggunakan kontrol akses berbutir halus, pengguna harus dipetakan ke `ultrawarm_manager` peran di OpenSearch Dasbor untuk melakukan panggilan API. UltraWarm

> **Note**
>
> `ultrawarm_manager`Peran mungkin tidak ditentukan pada beberapa domain OpenSearch Layanan yang sudah ada sebelumnya. Jika Anda tidak melihat peran di Dasbor, Anda harus membuatnya secara manual.

Konfigurasi izin

Jika Anda UltraWarm mengaktifkan domain OpenSearch Layanan yang sudah ada sebelumnya, `ultrawarm_manager` peran tersebut mungkin tidak ditentukan pada domain. Pengguna non-admin harus dipetakan ke peran ini untuk mengelola indeks hangat pada domain menggunakan kontrol akses berbutir halus. Untuk membuat secara manual peran `ultrawarm_manager`, lakukan langkah-langkah berikut:

1. Di OpenSearch Dasbor, buka Keamanan dan pilih Izin.
2. Pilih Buat grup tindakan dan konfigurasi grup-grup berikut:

Nama grup	Izin
`ultrawarm_cluster`	• `cluster:admin/ultrawarm/migration/list` • `cluster:monitor/nodes/stats`
`ultrawarm_index_read`	• `indices:admin/ultrawarm/migration/get` • `indices:admin/get`
`ultrawarm_index_write`	• `indices:admin/ultrawarm/migration/warm` • `indices:admin/ultrawarm/migration/hot` • `indices:monitor/stats` • `indices:admin/ultrawarm/migration/cancel`

3. Pilih Peran dan Buat peran.
4. Beri nama peran ultrawarm_manager.
5. Untuk Izin klaster, pilih `ultrawarm_cluster` dan `cluster_monitor`.
6. Untuk Indeks, ketik *.

7. Untuk Izin indeks, pilih `ultrawarm_index_read`, `ultrawarm_index_write`, dan `indices_monitor`.
8. Pilih Buat.
9. Setelah Anda membuat peran, petakan ke setiap pengguna atau peran backend yang akan mengelola UltraWarm indeks.

UltraWarm persyaratan penyimpanan dan pertimbangan kinerja

Sebagaimana tercakup dalamthe section called “Menghitung persyaratan penyimpanan”, data dalam penyimpanan panas menimbulkan overhead yang signifikan: replika, ruang cadangan Linux, dan OpenSearch ruang cadangan Layanan. Misalnya, pecahan primer 20 GiB dengan satu pecahan replika membutuhkan sekitar 58 GiB penyimpanan panas.

Karena menggunakan Amazon S3, UltraWarm tidak menimbulkan overhead ini. Saat menghitung persyaratan UltraWarm penyimpanan, Anda hanya mempertimbangkan ukuran pecahan utama. Daya tahan data di S3 menghilangkan kebutuhan untuk replika, dan S3 mengabstraksi setiap pertimbangan sistem operasi atau layanan. Serpihan 20 GiB yang sama membutuhkan 20 GiB penyimpanan hangat. Jika Anda menyediakan instans `ultrawarm1.large.search`, Anda dapat menggunakan semua 20 TiB dari penyimpanan maksimumnya untuk serpihan utama. Lihat the section called “UltraWarm kuota penyimpanan” untuk ringkasan tipe instans dan jumlah penyimpanan maksimum yang dapat ditangani masing-masing.

Dengan UltraWarm, kami masih merekomendasikan ukuran pecahan maksimum 50 GiB. Jumlah inti CPU dan jumlah RAM yang dialokasikan untuk setiap jenis UltraWarm instans memberi Anda gambaran tentang jumlah pecahan yang dapat mereka cari secara bersamaan. Perhatikan bahwa meskipun hanya pecahan primer yang dihitung terhadap UltraWarm penyimpanan di S3, OpenSearch Dasbor dan `_cat/indices` masih melaporkan ukuran UltraWarm indeks sebagai total semua pecahan primer dan replika.

Sebagai contoh, setiap instans `ultrawarm1.medium.search` memiliki dua core CPU dan dapat menangani hingga 1,5 TiB penyimpanan di S3. Dua dari instans ini memiliki gabungan 3 TiB penyimpanan, yang bekerja untuk sekitar 62 serpihan jika setiap serpihan berukuran 50 GiB. Jika permintaan ke klaster hanya mencari empat serpihan ini, performanya mungkin sangat baik. Jika permintaannya luas dan mencari semua 62 darinya, empat core CPU mungkin kesulitan untuk melakukan operasi. Pantau `WarmCPUUtilization` dan `WarmJVMMemoryPressure` UltraWarm metrik untuk memahami cara instans menangani beban kerja Anda.

Jika pencarian Anda luas atau sering, pertimbangkan untuk meninggalkan indeks dalam penyimpanan panas. Sama seperti OpenSearch beban kerja lainnya, langkah terpenting untuk menentukan apakah UltraWarm memenuhi kebutuhan Anda adalah melakukan pengujian klien yang representatif menggunakan kumpulan data yang realistis.

UltraWarm harga

Dengan penyimpanan panas, Anda membayar untuk apa yang Anda sediakan. Beberapa instans memerlukan volume Amazon EBS terlampir, sementara yang lain menyertakan penyimpanan instans. Apakah penyimpanan itu kosong atau penuh, Anda membayar harga yang sama.

Dengan UltraWarm penyimpanan, Anda membayar untuk apa yang Anda gunakan. Instans `ultrawarm1.large.search` dapat menangani hingga 20 TiB penyimpanan di S3, tetapi jika Anda hanya menyimpan 1 TiB data, Anda hanya akan ditagih untuk 1 TiB data. Seperti semua jenis node lainnya, Anda juga membayar tarif per jam untuk setiap UltraWarm node. Untuk informasi selengkapnya, lihat the section called “Harga untuk Amazon OpenSearch Service”.

Mengaktifkan UltraWarm

Konsol adalah cara paling mudah untuk membuat domain yang menggunakan penyimpanan hangat. Saat membuat domain, pilih Aktifkan node UltraWarm data dan jumlah node hangat yang Anda inginkan. Proses dasar yang sama bekerja pada domain yang ada, asalkan proses tersebut memenuhi prasyarat. Bahkan setelah status domain berubah dari Processing ke Active, UltraWarm mungkin tidak tersedia untuk digunakan selama beberapa jam.

Anda juga dapat menggunakan API konfigurasi AWS CLIatau untuk mengaktifkan UltraWarm, khususnya `WarmEnabled`, `WarmCount`, dan `WarmType` opsi di`ClusterConfig`.

> **Note**
>
> Domain mendukung jumlah maksimum simpul hangat. Untuk detailnya, lihat the section called “Kuota”.

Perintah CLI sampel

Perintah AWS CLI berikut membuat domain dengan tiga simpul data, tiga simpul utama terdedikasi, enam simpul hangat, dan kontrol akses detail diaktifkan:

```
aws opensearch create-domain \
  --domain-name my-domain \
  --engine-version Opensearch_1.0 \
  --cluster-config
 InstanceCount=3,InstanceType=r6g.large.search,DedicatedMasterEnabled=true,DedicatedMasterType=
 \
  --ebs-options EBSEnabled=true,VolumeType=gp2,VolumeSize=11 \
  --node-to-node-encryption-options Enabled=true \
  --encryption-at-rest-options Enabled=true \
  --domain-endpoint-options EnforceHTTPS=true,TLSSecurityPolicy=Policy-Min-
TLS-1-2-2019-07 \
  --advanced-security-options
 Enabled=true,InternalUserDatabaseEnabled=true,MasterUserOptions='{MasterUserName=master-
user,MasterUserPassword=master-password}' \
  --access-policies '{"Version":"2012-10-17","Statement":
[{"Effect":"Allow","Principal":{"AWS":["123456789012"]},"Action":
["es:*"],"Resource":"arn:aws:es:us-west-1:123456789012:domain/my-domain/*"}]}' \
  --region us-east-1
```

Untuk informasi detail, lihat Referensi Perintah AWS CLI.

Sampel permintaan API konfigurasi

Permintaan berikut ke API konfigurasi membuat domain dengan tiga simpul data, tiga simpul utama terdedikasi, enam simpul hangat, dan kontrol akses detail diaktifkan dan kebijakan akses terbatas:

```
POST https://es.us-east-2.amazonaws.com/2021-01-01/opensearch/domain
{
  "ClusterConfig": {
    "InstanceCount": 3,
    "InstanceType": "r6g.large.search",
    "DedicatedMasterEnabled": true,
    "DedicatedMasterType": "r6g.large.search",
    "DedicatedMasterCount": 3,
    "ZoneAwarenessEnabled": true,
    "ZoneAwarenessConfig": {
      "AvailabilityZoneCount": 3
    },
    "WarmEnabled": true,
    "WarmCount": 6,
    "WarmType": "ultrawarm1.medium.search"
  },
  "EBSOptions": {
```

```
    "EBSEnabled": true,
    "VolumeType": "gp2",
    "VolumeSize": 11
  },
  "EncryptionAtRestOptions": {
    "Enabled": true
  },
  "NodeToNodeEncryptionOptions": {
    "Enabled": true
  },
  "DomainEndpointOptions": {
    "EnforceHTTPS": true,
    "TLSSecurityPolicy": "Policy-Min-TLS-1-2-2019-07"
  },
   "AdvancedSecurityOptions": {
    "Enabled": true,
    "InternalUserDatabaseEnabled": true,
    "MasterUserOptions": {
      "MasterUserName": "master-user",
      "MasterUserPassword": "master-password"
    }
  },
  "EngineVersion": "Opensearch_1.0",
  "DomainName": "my-domain",
  "AccessPolicies": "{\"Version\":\"2012-10-17\",\"Statement\":[{\"Effect\":\"Allow
\",\"Principal\":{\"AWS\":[\"123456789012\"]},\"Action\":[\"es:*\"],\"Resource\":
\"arn:aws:es:us-east-1:123456789012:domain/my-domain/*\"}]}"
}
```

Untuk informasi selengkapnya, lihat Referensi API OpenSearch Layanan Amazon.

Migrasi indeks ke penyimpanan UltraWarm

Jika Anda selesai menulis ke indeks dan tidak lagi membutuhkan performa penelusuran tercepat, migrasikan dari panas ke UltraWarm:

```
POST _ultrawarm/migration/my-index/_warm
```

Kemudian periksa status migrasi:

```
GET _ultrawarm/migration/my-index/_status
```

```
{
  "migration_status": {
    "index": "my-index",
    "state": "RUNNING_SHARD_RELOCATION",
    "migration_type": "HOT_TO_WARM",
    "shard_level_status": {
      "running": 0,
      "total": 5,
      "pending": 3,
      "failed": 0,
      "succeeded": 2
    }
  }
}
```

Kesehatan indeks harus hijau untuk melakukan migrasi. Jika Anda memigrasikan beberapa indeks secara berurutan, Anda bisa mendapatkan ringkasan semua migrasi dalam plaintext, mirip dengan API: `_cat`

```
GET _ultrawarm/migration/_status?v

index    migration_type state
my-index HOT_TO_WARM    RUNNING_SHARD_RELOCATION
```

OpenSearch Layanan memigrasikan satu indeks pada satu waktu ke UltraWarm. Anda dapat memiliki hingga 200 migrasi dalam antrian. Setiap permintaan yang melebihi batas akan ditolak. Untuk memeriksa jumlah migrasi saat ini dalam antrean, pantau metrik `HotToWarmMigrationQueueSize`. Indeks tetap tersedia selama proses migrasi — tidak ada waktu henti.

Proses migrasi memiliki status sebagai berikut:

```
PENDING_INCREMENTAL_SNAPSHOT
RUNNING_INCREMENTAL_SNAPSHOT
FAILED_INCREMENTAL_SNAPSHOT
PENDING_FORCE_MERGE
RUNNING_FORCE_MERGE
FAILED_FORCE_MERGE
PENDING_FULL_SNAPSHOT
RUNNING_FULL_SNAPSHOT
FAILED_FULL_SNAPSHOT
```

```
PENDING_SHARD_RELOCATION
RUNNING_SHARD_RELOCATION
FINISHED_SHARD_RELOCATION
```

Seperti yang ditunjukkan oleh status ini, migrasi mungkin gagal selama snapshot, relokasi serpihan, atau penggabungan paksa. Kegagalan selama snapshot atau relokasi serpihan biasanya karena kegagalan simpul atau masalah konektivitas S3. Kurangnya ruang disk biasanya menjadi penyebab kegagalan penggabungan paksa.

Setelah migrasi selesai, permintaan `_status` yang sama mengembalikan kesalahan. Jika Anda memeriksa indeks pada saat itu, Anda dapat melihat beberapa pengaturan yang unik untuk indeks hangat:

```
GET my-index/_settings

{
  "my-index": {
    "settings": {
      "index": {
        "refresh_interval": "-1",
        "auto_expand_replicas": "false",
        "provided_name": "my-index",
        "creation_date": "1599241458998",
        "unassigned": {
          "node_left": {
            "delayed_timeout": "5m"
          }
        },
        "number_of_replicas": "1",
        "uuid": "GswyCdR0RSq0SJYmzsIpiw",
        "version": {
          "created": "7070099"
        },
        "routing": {
          "allocation": {
            "require": {
              "box_type": "warm"
            }
          }
        },
        "number_of_shards": "5",
        "merge": {
          "policy": {
```

```
            "max_merge_at_once_explicit": "50"
          }
        }
      }
    }
  }
}
```

- `number_of_replicas`, dalam kasus ini, adalah jumlah replika pasif, yang tidak mengonsumsi ruang disk.
- `routing.allocation.require.box_type` menetapkan bahwa indeks harus menggunakan simpul hangat daripada simpul data standar.
- `merge.policy.max_merge_at_once_explicit` menentukan jumlah segmen untuk secara bersamaan digabungkan selama migrasi.

Indeks dalam penyimpanan hangat hanya dapat dibaca kecuali Anda mengembalikannya ke penyimpanan panas, yang UltraWarm paling cocok untuk data yang tidak dapat diubah, seperti log. Anda dapat menanyakan indeks dan menghapusnya, tetapi Anda tidak dapat menambahkan, memperbarui, atau menghapus dokumen individual. Jika Anda mencobanya, Anda mungkin mengalami kesalahan berikut:

```
{
  "error" : {
    "root_cause" : [
      {
        "type" : "cluster_block_exception",
        "reason" : "index [indexname] blocked by: [TOO_MANY_REQUESTS/12/disk usage
 exceeded flood-stage watermark, index has read-only-allow-delete block];"
      }
    ],
    "type" : "cluster_block_exception",
    "reason" : "index [indexname] blocked by: [TOO_MANY_REQUESTS/12/disk usage exceeded
 flood-stage watermark, index has read-only-allow-delete block];"
  },
  "status" : 429
}
```

Mengotomatisasi migrasi

Sebaiknya gunakan the section called “Manajemen state indeks” untuk mengotomatisasi proses migrasi setelah indeks mencapai usia tertentu atau memenuhi kondisi lain. Lihat kebijakan sampel yang menunjukkan alur kerja ini.

Penyetelan migrasi

Migrasi indeks ke UltraWarm penyimpanan memerlukan penggabungan paksa. Setiap OpenSearch indeks terdiri dari sejumlah pecahan, dan setiap pecahan terdiri dari beberapa segmen Lucene. Operasi penggabungan paksa membersihkan dokumen yang ditandai untuk penghapusan dan menghemat ruang disk. Secara default, UltraWarm menggabungkan indeks menjadi satu segmen.

Anda dapat mengubah nilai ini hingga 1.000 segmen menggunakan pengaturan `index.ultrawarm.migration.force_merge.max_num_segments`. Nilai yang lebih tinggi mempercepat proses migrasi, tetapi meningkatkan latensi kueri untuk indeks hangat setelah migrasi selesai. Untuk mengubah pengaturan, buat permintaan berikut:

```
PUT my-index/_settings
{
  "index": {
    "ultrawarm": {
      "migration": {
        "force_merge": {
          "max_num_segments": 1
        }
      }
    }
  }
}
```

Untuk memeriksa berapa lama tahap proses migrasi ini berlangsung, pantau `HotToWarmMigrationForceMergeLatency` metrik.

Membatalkan migrasi

UltraWarm menangani migrasi secara berurutan, dalam antrian. Jika migrasi dalam antrean, namun belum dimulai, Anda dapat menghapusnya dari antrean menggunakan permintaan berikut:

```
POST _ultrawarm/migration/_cancel/my-index
```

Jika domain Anda menggunakan kontrol akses detail, Anda harus memiliki izin `indices:admin/ultrawarm/migration/cancel` untuk membuat permintaan ini.

Daftar indeks panas dan hangat

UltraWarm menambahkan dua opsi tambahan, mirip dengan`_all`, untuk membantu mengelola indeks panas dan hangat. Untuk daftar semua indeks hangat atau panas, buat permintaan berikut:

```
GET _warm
GET _hot
```

Anda dapat menggunakan opsi ini dalam permintaan lain yang menentukan indeks, seperti:

```
_cat/indices/_warm
_cluster/state/_all/_hot
```

Mengembalikan indeks hangat ke penyimpanan panas

Jika Anda perlu menulis ke indeks lagi, migrasikan kembali ke penyimpanan panas:

```
POST _ultrawarm/migration/my-index/_hot
```

Anda dapat memiliki hingga 10 migrasi antrian dari penyimpanan hangat ke penyimpanan panas sekaligus. OpenSearch Layanan memproses permintaan migrasi satu per satu, dalam urutan bahwa mereka mengantri. Untuk memeriksa jumlah saat ini, pantau metrik `WarmToHotMigrationQueueSize`.

Setelah migrasi selesai, periksa pengaturan indeks untuk memastikan itu memenuhi kebutuhan Anda. Indeks kembali ke penyimpanan panas dengan satu replika.

Memulihkan indeks hangat dari snapshot

Selain repositori standar untuk snapshot otomatis, UltraWarm tambahkan repositori kedua untuk indeks hangat,. `cs-ultrawarm` Setiap snapshot dalam repositori ini hanya berisi satu indeks. Jika Anda menghapus indeks hangat, snapshotnya tetap berada di repositori `cs-ultrawarm` selama 14 hari, sama seperti snapshot otomatis lainnya.

Saat Anda memulihkan snapshot dari `cs-ultrawarm`, itu memulihkan ke penyimpanan hangat, bukan penyimpanan panas. Snapshot di repositori `cs-automated` dan `cs-automated-enc` memulihkan ke penyimpanan panas.

Untuk mengembalikan UltraWarm snapshot ke penyimpanan hangat

1. Identifikasi snapshot terbaru yang berisi indeks yang ingin Anda pulihkan:

```
GET _snapshot/cs-ultrawarm/_all?verbose=false

{
  "snapshots": [{
    "snapshot": "snapshot-name",
    "version": "1.0",
    "indices": [
      "my-index"
    ]
  }]
}
```

> **Note**
>
> Secara default, `GET _snapshot/<repo>` operasi menampilkan informasi data verbose seperti waktu mulai, waktu akhir, dan durasi untuk setiap snapshot dalam repositori. `GET _snapshot/<repo>`Operasi mengambil informasi dari file setiap snapshot yang terdapat dalam repositori. Jika Anda tidak memerlukan waktu mulai, waktu akhir, dan durasi dan hanya memerlukan informasi nama dan indeks snapshot, sebaiknya gunakan `verbose=false` parameter saat mencantumkan snapshot untuk meminimalkan waktu pemrosesan dan mencegah waktu habis.

2. Jika indeks sudah ada, hapuslah:

```
DELETE my-index
```

Jika Anda tidak ingin menghapus indeks, kembalikan indeks ke penyimpanan panas dan indeks ulang itu.

3. Memulihkan snapshot:

```
POST _snapshot/cs-ultrawarm/snapshot-name/_restore
```

UltraWarm mengabaikan pengaturan indeks apa pun yang Anda tentukan dalam permintaan pemulihan ini, tetapi Anda dapat menentukan opsi seperti `rename_pattern`

dan`rename_replacement`. Untuk ringkasan opsi pemulihan OpenSearch snapshot, lihat OpenSearch dokumentasi.

Cuplikan manual dari indeks hangat

Anda dapat mengambil snapshot manual dari indeks hangat, tetapi kami tidak merekomendasikannya. Repositori `cs-ultrawarm` otomatis sudah berisi snapshot untuk setiap indeks hangat, yang diambil selama migrasi, tanpa biaya tambahan.

Secara default, OpenSearch Layanan tidak menyertakan indeks hangat dalam snapshot manual. Misalnya, panggilan berikut hanya menyertakan indeks panas:

```
PUT _snapshot/my-repository/my-snapshot
```

Jika Anda memilih untuk mengambil snapshot manual dari indeks hangat, beberapa pertimbangan penting berlaku.

- Anda tidak dapat mencampur indeks panas dan hangat. Sebagai contoh, permintaan berikut gagal:

  ```
  PUT _snapshot/my-repository/my-snapshot
  {
    "indices": "warm-index-1,hot-index-1",
    "include_global_state": false
  }
  ```

  Jika mereka menyertakan campuran indeks panas dan hangat, pernyataan wildcard (*) juga gagal.

- Anda hanya dapat menyertakan satu indeks hangat per snapshot. Sebagai contoh, permintaan berikut gagal:

  ```
  PUT _snapshot/my-repository/my-snapshot
  {
    "indices": "warm-index-1,warm-index-2,other-warm-indices-*",
    "include_global_state": false
  }
  ```

  Permintaan ini berhasil:

  ```
  PUT _snapshot/my-repository/my-snapshot
  {
  ```

```
  "indices": "warm-index-1",
  "include_global_state": false
}
```

- Snapshot manual selalu mengembalikan ke penyimpanan panas, bahkan jika awalnya termasuk indeks hangat.

Migrasi indeks hangat ke cold storage

Jika Anda memiliki data UltraWarm yang jarang Anda kueri, pertimbangkan untuk memigrasikannya ke penyimpanan dingin. Penyimpanan dingin dimaksudkan untuk data yang hanya Anda akses sesekali atau tidak lagi digunakan secara aktif. Anda tidak dapat membaca dari atau menulis ke indeks dingin, tetapi Anda dapat memigrasikannya kembali ke penyimpanan hangat tanpa biaya kapan pun Anda perlu menanyakannya. Untuk petunjuk, lihat the section called “Migrasi indeks ke cold storage”.

Menonaktifkan UltraWarm

Konsol adalah cara paling sederhana untuk menonaktifkan UltraWarm. Pilih domain, Tindakan, dan Edit konfigurasi cluster. Hapus pilihan Aktifkan node UltraWarm data dan pilih Simpan perubahan. Anda juga dapat menggunakan opsi `WarmEnabled` di AWS CLI dan API konfigurasi.

Sebelum menonaktifkan UltraWarm, Anda harus menghapus semua indeks hangat atau memigrasikannya kembali ke penyimpanan panas. Setelah penyimpanan hangat kosong, tunggu lima menit sebelum mencoba menonaktifkan UltraWarm.

Penyimpanan dingin untuk OpenSearch Layanan Amazon

Penyimpanan dingin memungkinkan Anda menyimpan sejumlah data yang jarang diakses atau historis di domain OpenSearch Layanan Amazon Anda dan menganalisisnya sesuai permintaan, dengan biaya lebih rendah daripada tingkatan penyimpanan lainnya. Penyimpanan dingin sesuai jika Anda perlu melakukan penelitian berkala atau analisis forensik pada data lama Anda. Contoh praktis data yang cocok untuk penyimpanan dingin termasuk log yang jarang diakses, data yang harus dipelihara untuk memenuhi persyaratan kepatuhan, atau log yang memiliki nilai historis.

Mirip dengan UltraWarmpenyimpanan, penyimpanan dingin didukung oleh Amazon S3. Saat Anda perlu menanyakan data dingin, Anda dapat secara selektif melampirkannya ke UltraWarm node yang ada. Anda dapat mengelola migrasi dan siklus hidup data dingin Anda secara manual atau dengan kebijakan Index State Management.

Topik

Prasyarat

Penyimpanan dingin memiliki prasyarat berikut:

- Cold storage membutuhkan OpenSearch atau Elasticsearch versi 7.9 atau yang lebih baru.
- Untuk mengaktifkan penyimpanan dingin pada domain OpenSearch Layanan, Anda juga harus mengaktifkan UltraWarm pada domain yang sama.
- Untuk menggunakan penyimpanan dingin, domain harus memiliki simpul utama khusus.
- Jika domain Anda menggunakan tipe instans T2 atau T3 untuk simpul data, Anda tidak dapat menggunakan penyimpanan dingin.
- Jika indeks Anda menggunakan codec kompresi Zstandard (`"index.codec": "zstd"`atau`"index.codec": "zstd_no_dict"`), Anda tidak dapat memindahkannya ke penyimpanan dingin.
- Jika indeks Anda menggunakan perkiraan k-nn (`"index.knn": true`), Anda tidak dapat memindahkannya ke cold storage.

- Jika domain menggunakan kontrol akses berbutir halus, pengguna non-admin harus dipetakan ke `cold_manager` peran di OpenSearch Dasbor untuk mengelola indeks dingin.

> **Note**
>
> `cold_manager`Peran tersebut mungkin tidak ada di beberapa domain OpenSearch Layanan yang sudah ada sebelumnya. Jika Anda tidak melihat peran di Dasbor, Anda harus membuatnya secara manual.

Konfigurasi izin

Jika Anda mengaktifkan penyimpanan dingin pada domain OpenSearch Layanan yang sudah ada sebelumnya, `cold_manager` peran tersebut mungkin tidak ditentukan pada domain. Jika domain menggunakan kontrol akses berbutir halus, pengguna non-admin harus dipetakan ke peran ini untuk mengelola indeks dingin. Untuk membuat secara manual peran `cold_manager`, lakukan langkah-langkah berikut:

1. Di OpenSearch Dasbor, buka Keamanan dan pilih Izin.
2. Pilih Buat grup tindakan dan konfigurasi grup-grup berikut:

Nama grup	Izin
`cold_cluster`	• `cluster:monitor/nodes/stats` • `cluster:admin/ultrawarm*` • `cluster:admin/cold/*`
`cold_index`	• `indices:monitor/stats` • `indices:data/read/minmax` • `indices:admin/ultrawarm/migration/get` • `indices:admin/ultrawarm/migration/cancel`

3. Pilih Peran dan Buat peran.
4. Nama peran cold_manager.
5. Untuk izin Cluster, pilih `cold_cluster` grup yang Anda buat.
6. Untuk Indeks, masukkan `*`.

7. Untuk Izin indeks, pilih grup `cold_index` yang Anda buat.
8. Pilih Buat.
9. Setelah Anda membuat peran, petakan ke setiap pengguna atau peran backend yang mengelola indeks dingin.

Persyaratan penyimpanan UltraWarm dan pertimbangan performa

Karena penyimpanan dingin menggunakan Amazon S3, penyimpanan dingin tidak menimbulkan overhead penyimpanan panas, seperti replika, ruang cadangan Linux, dan ruang cadangan Layanan. OpenSearch Penyimpanan dingin tidak memiliki tipe instans tertentu karena tidak memiliki kapasitas komputasi yang melekat padanya. Anda dapat menyimpan sejumlah data dalam penyimpanan dingin. Pantau `ColdStorageSpaceUtilization` metrik di Amazon CloudWatch untuk melihat berapa banyak ruang penyimpanan dingin yang Anda gunakan.

Harga penyimpanan dingin

Mirip dengan UltraWarm penyimpanan, dengan cold storage Anda hanya membayar untuk penyimpanan data. Tidak ada biaya komputasi untuk data dingin dan Anda tidak akan ditagih jika tidak ada data dalam penyimpanan data.

Anda tidak dikenakan biaya transfer saat memindahkan data antara penyimpanan dingin dan hangat. Sementara indeks sedang bermigrasi antara penyimpanan hangat dan dingin, Anda terus membayar hanya satu salinan indeks. Setelah migrasi selesai, indeks ditagih sesuai dengan tingkat penyimpanan yang dimigrasikan. Untuk informasi selengkapnya tentang harga cold storage, lihat harga Amazon OpenSearch Service.

Mengaktifkan penyimpanan dingin

Konsol adalah cara paling mudah untuk membuat domain yang menggunakan penyimpanan dingin. Saat membuat domain, pilih Aktifkan penyimpanan dingin. Proses yang sama bekerja pada domain yang ada selama Anda memenuhi prasyarat. Bahkan setelah status domain berubah dari Pemrosesan ke Aktif, penyimpanan dingin mungkin tidak tersedia selama beberapa jam.

Anda juga dapat menggunakan AWS CLI atau API konfigurasi untuk mengaktifkan penyimpanan dingin.

Contoh perintah CLI

Perintah AWS CLI berikut membuat domain dengan tiga simpul data, tiga simpul utama khusus, penyimpanan dingin diaktifkan, dan kontrol akses detail diaktifkan:

```
aws opensearch create-domain \
  --domain-name my-domain \
  --engine-version Opensearch_1.0 \
  --cluster-
config ColdStorageOptions={Enabled=true},WarmEnabled=true,WarmCount=4,WarmType=ultrawarm1.mediu
 \
  --ebs-options EBSEnabled=true,VolumeType=gp2,VolumeSize=11 \
  --node-to-node-encryption-options Enabled=true \
  --encryption-at-rest-options Enabled=true \
  --domain-endpoint-options EnforceHTTPS=true,TLSSecurityPolicy=Policy-Min-
TLS-1-2-2019-07 \
  --advanced-security-options
 Enabled=true,InternalUserDatabaseEnabled=true,MasterUserOptions='{MasterUserName=master-
user,MasterUserPassword=master-password}' \
  --region us-east-2
```

Untuk informasi rinci, lihat AWS CLI Referensi Perintah.

Contoh permintaan API konfigurasi

Permintaan berikut ke API konfigurasi membuat domain dengan tiga node data, tiga node utama khusus, penyimpanan dingin diaktifkan, dan kontrol akses halus diaktifkan:

```
POST https://es.us-east-2.amazonaws.com/2021-01-01/opensearch/domain
{
  "ClusterConfig": {
    "InstanceCount": 3,
    "InstanceType": "r6g.large.search",
    "DedicatedMasterEnabled": true,
    "DedicatedMasterType": "r6g.large.search",
    "DedicatedMasterCount": 3,
    "ZoneAwarenessEnabled": true,
    "ZoneAwarenessConfig": {
      "AvailabilityZoneCount": 3
     },
    "WarmEnabled": true,
    "WarmCount": 4,
```

```
    "WarmType": "ultrawarm1.medium.search",
    "ColdStorageOptions": {
       "Enabled": true
     }
  },
  "EBSOptions": {
    "EBSEnabled": true,
    "VolumeType": "gp2",
    "VolumeSize": 11
  },
  "EncryptionAtRestOptions": {
    "Enabled": true
  },
  "NodeToNodeEncryptionOptions": {
    "Enabled": true
  },
  "DomainEndpointOptions": {
    "EnforceHTTPS": true,
    "TLSSecurityPolicy": "Policy-Min-TLS-1-2-2019-07"
  },
   "AdvancedSecurityOptions": {
    "Enabled": true,
    "InternalUserDatabaseEnabled": true,
    "MasterUserOptions": {
      "MasterUserName": "master-user",
      "MasterUserPassword": "master-password"
    }
  },
  "EngineVersion": "Opensearch_1.0",
  "DomainName": "my-domain"
}
```

Untuk informasi selengkapnya, lihat Referensi API OpenSearch Layanan Amazon.

Mengelola indeks dingin di Dasbor OpenSearch

Anda dapat mengelola indeks panas, hangat, dan dingin dengan antarmuka Dasbor yang ada di domain OpenSearch Layanan Anda. Dasbor memungkinkan Anda untuk memigrasikan indeks antara penyimpanan hangat dan dingin, dan memantau status migrasi indeks, tanpa menggunakan CLI atau API konfigurasi. Untuk informasi selengkapnya, lihat Mengelola indeks di OpenSearch Dasbor.

Migrasi indeks ke cold storage

Saat memigrasikan indeks ke penyimpanan dingin, Anda menyediakan rentang waktu agar data lebih mudah ditemukan. Anda dapat memilih bidang timestamp berdasarkan data dalam indeks Anda, secara manual memberikan awal dan akhir timestamp, atau memilih untuk tidak menentukannya.

Parameter	Nilai yang didukung	Deskripsi
`timestamp_field`	Bidang tanggal/waktu dari pemetaan indeks.	Nilai minimum dan maksimum bidang yang disediakan dihitung dan disimpan sebagai metadata `start_time` dan `end_time` untuk indeks dingin.
`start_time` dan `end_time`	Salah satu format berikut: • strict_date_optional_time. Misalnya `yyyy-MM-dd'T'HH:mm:ss.SSSZ` atau `yyyy-MM-dd` • Waktu jangka waktu dalam milidetik	Nilai yang disediakan disimpan sebagai metadata `start_time` dan `end_time` untuk indeks dingin.

Jika Anda tidak ingin menentukan timestamp, tambahkan `?ignore=timestamp` untuk permintaan sebagai gantinya.

Permintaan berikut memigrasikan indeks hangat ke penyimpanan dingin dan menyediakan waktu mulai dan akhir untuk data dalam indeks tersebut:

```
POST _ultrawarm/migration/my-index/_cold
  {
    "start_time": "2020-03-09",
    "end_time": "2020-03-09T23:00:00Z"
  }
```

Kemudian periksa status migrasi:

```
GET _ultrawarm/migration/my-index/_status

{
  "migration_status": {
    "index": "my-index",
    "state": "RUNNING_METADATA_RELOCATION",
    "migration_type": "WARM_TO_COLD"
  }
}
```

OpenSearch Layanan memigrasikan satu indeks pada satu waktu ke cold storage. Anda dapat memiliki hingga 100 migrasi dalam antrian. Setiap permintaan yang melebihi batas akan ditolak. Untuk memeriksa jumlah migrasi saat ini dalam antrean, pantau metrik `WarmToColdMigrationQueueSize`. Proses migrasi memiliki status sebagai berikut:

```
ACCEPTED_COLD_MIGRATION - Migration request is accepted and queued.
RUNNING_METADATA_MIGRATION - The migration request was selected for execution and
 metadata is migrating to cold storage.
FAILED_METADATA_MIGRATION - The attempt to add index metadata has failed and all
 retries are exhausted.
PENDING_INDEX_DETACH - Index metadata migration to cold storage is completed. Preparing
 to detach the warm index state from the local cluster.
RUNNING_INDEX_DETACH - Local warm index state from the cluster is being removed. Upon
 success, the migration request will be completed.
FAILED_INDEX_DETACH - The index detach process failed and all retries are exhausted.
```

Mengotomatiskan perpindahan ke penyimpanan dingin

Anda dapat menggunakan Indeks State Management untuk mengotomatisasi proses migrasi setelah indeks mencapai usia tertentu atau memenuhi kondisi lain. Lihat kebijakan sampel, yang menunjukkan cara memigrasikan indeks secara otomatis dari penyimpanan panas UltraWarm ke penyimpanan dingin.

> **Note**
>
> Eksplisit `timestamp_field` diperlukan untuk memindahkan indeks ke penyimpanan dingin menggunakan kebijakan Manajemen Negara Indeks.

Membatalkan migrasi ke penyimpanan dingin

Jika migrasi ke penyimpanan dingin dalam antrian atau dalam keadaan gagal, Anda dapat membatalkan migrasi menggunakan permintaan berikut:

```
POST _ultrawarm/migration/_cancel/my-index

{
  "acknowledged" : true
}
```

Jika domain Anda menggunakan kontrol akses detail, Anda memerlukan izin `indices:admin/ultrawarm/migration/cancel` untuk membuat permintaan ini.

Daftar indeks dingin

Sebelum melakukan kueri, Anda dapat membuat daftar indeks di cold storage untuk memutuskan indeks mana yang akan dimigrasi UltraWarm untuk analisis lebih lanjut. Permintaan berikut mencantumkan semua indeks dingin, diurutkan berdasarkan nama indeks:

```
GET _cold/indices/_search
```

Sampel respon

```
{
  "pagination_id" : "je7MtGbClwBF/2Zp9Utk/h3yCo8nvbEXAMPLEKEY",
  "total_results" : 3,
  "indices" : [
    {
      "index" : "my-index-1",
      "index_cold_uuid" : "hjEoh26mRRCFxRIMdgvLmg",
      "size" : 10339,
      "creation_date" : "2021-06-28T20:23:31.206Z",
      "start_time" : "2020-03-09T00:00Z",
      "end_time" : "2020-03-09T23:00Z"
    },
    {
      "index" : "my-index-2",
      "index_cold_uuid" : "0vIS2n-oROmOWDFmwFIgdw",
      "size" : 6068,
      "creation_date" : "2021-07-15T19:41:18.046Z",
```

```
      "start_time" : "2020-03-09T00:00Z",
      "end_time" : "2020-03-09T23:00Z"
    },
    {
      "index" : "my-index-3",
      "index_cold_uuid" : "EaeXOBodTLiDYcivKsXVLQ",
      "size" : 32403,
      "creation_date" : "2021-07-08T00:12:01.523Z",
      "start_time" : "2020-03-09T00:00Z",
      "end_time" : "2020-03-09T23:00Z"
    }
  ]
}
```

Penyaringan

Anda dapat memfilter indeks dingin berdasarkan pola indeks berbasis awalan dan offset rentang waktu.

Permintaan berikut mencantumkan indeks yang cocok dengan pola awalan: `event-*`

```
GET _cold/indices/_search
 {
   "filters":{
      "index_pattern": "event-*"
   }
 }
```

Sampel respon

```
{
  "pagination_id" : "je7MtGbClwBF/2Zp9Utk/h3yCo8nvbEXAMPLEKEY",
  "total_results" : 1,
  "indices" : [
    {
      "index" : "events-index",
      "index_cold_uuid" : "4eFiab7rRfSvp3slrIsIKA",
      "size" : 32263273,
      "creation_date" : "2021-08-18T18:25:31.845Z",
      "start_time" : "2020-03-09T00:00Z",
      "end_time" : "2020-03-09T23:00Z"
    }
  ]
```

```
}
```

Permintaan berikut mengembalikan indeks dengan `start_time` dan bidang `end_time` metadata antara dan: `2019-03-01` `2020-03-01`

```
GET _cold/indices/_search
{
  "filters": {
    "time_range": {
      "start_time": "2019-03-01",
      "end_time": "2020-03-01"
    }
  }
}
```

Sampel respon

```
{
  "pagination_id" : "je7MtGbClwBF/2Zp9Utk/h3yCo8nvbEXAMPLEKEY",
  "total_results" : 1,
  "indices" : [
    {
      "index" : "my-index",
      "index_cold_uuid" : "4eFiab7rRfSvp3slrIsIKA",
      "size" : 32263273,
      "creation_date" : "2021-08-18T18:25:31.845Z",
      "start_time" : "2019-05-09T00:00Z",
      "end_time" : "2019-09-09T23:00Z"
    }
  ]
}
```

Penyortiran

Anda dapat mengurutkan indeks dingin berdasarkan bidang metadata seperti nama indeks atau ukuran. Permintaan berikut mencantumkan semua indeks yang diurutkan berdasarkan ukuran dalam urutan menurun:

```
GET _cold/indices/_search
 {
 "sort_key": "size:desc"
```

```
 }
```

Sampel respon

```
{
  "pagination_id" : "je7MtGbClwBF/2Zp9Utk/h3yCo8nvbEXAMPLEKEY",
  "total_results" : 5,
  "indices" : [
    {
      "index" : "my-index-6",
      "index_cold_uuid" : "4eFiab7rRfSvp3slrIsIKA",
      "size" : 32263273,
      "creation_date" : "2021-08-18T18:25:31.845Z",
      "start_time" : "2020-03-09T00:00Z",
      "end_time" : "2020-03-09T23:00Z"
    },
    {
      "index" : "my-index-9",
      "index_cold_uuid" : "mbD3ZRVDRI6ONqgEOsJyUA",
      "size" : 57922,
      "creation_date" : "2021-07-07T23:41:35.640Z",
      "start_time" : "2020-03-09T00:00Z",
      "end_time" : "2020-03-09T23:00Z"
    },
    {
      "index" : "my-index-5",
      "index_cold_uuid" : "EaeXOBodTLiDYcivKsXVLQ",
      "size" : 32403,
      "creation_date" : "2021-07-08T00:12:01.523Z",
      "start_time" : "2020-03-09T00:00Z",
      "end_time" : "2020-03-09T23:00Z"
    }
  ]
}
```

Kunci pengurutan valid lainnya adalah `start_time:asc/desc`, `end_time:asc/desc`, dan `index_name:asc/desc`.

Paginasi

Anda dapat membuat halaman daftar indeks dingin. Konfigurasikan jumlah indeks yang akan dikembalikan per halaman dengan `page_size` parameter (defaultnya adalah 10). Setiap `_search`

permintaan pada indeks dingin Anda mengembalikan permintaan `pagination_id` yang dapat Anda gunakan untuk panggilan berikutnya.

Permintaan berikut memberi halaman atas hasil `_search` permintaan indeks dingin Anda dan menampilkan 100 hasil berikutnya:

```
GET _cold/indices/_search?page_size=100
{
"pagination_id": "je7MtGbClwBF/2Zp9Utk/h3yCo8nvbEXAMPLEKEY"
}
```

Migrasi indeks dingin ke penyimpanan hangat

Setelah Anda mempersempit daftar indeks dingin dengan kriteria pemfilteran di bagian sebelumnya, migrasikan kembali ke UltraWarm tempat Anda dapat menanyakan data dan menggunakannya untuk membuat visualisasi.

Permintaan berikut memigrasikan dua indeks dingin kembali ke penyimpanan hangat:

```
POST _cold/migration/_warm
 {
 "indices": "my-index1,my-index2"
 }


{
  "acknowledged" : true
}
```

Untuk memeriksa status migrasi dan mengambil ID migrasi, kirim permintaan berikut:

```
GET _cold/migration/_status
```

Sampel respon

```
{
  "cold_to_warm_migration_status" : [
    {
      "migration_id" : "tyLjXCA-S76zPQbPVHkOKA",
      "indices" : [
        "my-index1,my-index2"
```

```
      ],
      "state" : "RUNNING_INDEX_CREATION"
    }
  ]
}
```

Untuk mendapatkan informasi migrasi spesifik indeks, termasuk nama indeks:

```
GET _cold/migration/my-index/_status
```

Daripada menentukan indeks, Anda dapat membuat daftar indeks berdasarkan status migrasi mereka saat ini. Nilai yang valid adalah `_failed`, `_accepted`, dan `_all`.

Perintah berikut mendapatkan status semua indeks dalam satu permintaan migrasi:

```
GET _cold/migration/_status?migration_id=my-migration-id
```

Mengambil ID migrasi menggunakan permintaan status. Untuk informasi migrasi terperinci, tambahkan `&verbose=true`.

Anda dapat memigrasikan indeks dari penyimpanan dingin ke penyimpanan hangat dalam batch 10 atau kurang, dengan maksimum 100 indeks dimigrasikan secara bersamaan. Setiap permintaan yang melebihi batas akan ditolak. Untuk memeriksa jumlah migrasi saat ini yang sedang berlangsung, pantau `ColdToWarmMigrationQueueSize` metriknya. Proses migrasi memiliki status sebagai berikut:

```
ACCEPTED_MIGRATION_REQUEST - Migration request is accepted and queued.
RUNNING_INDEX_CREATION - Migration request is picked up for processing and will create
 warm indexes in the cluster.
PENDING_COLD_METADATA_CLEANUP - Warm index is created and the migration service will
 attempt to clean up cold metadata.
RUNNING_COLD_METADATA_CLEANUP - Cleaning up cold metadata from the indexes migrated to
 warm storage.
FAILED_COLD_METADATA_CLEANUP - Failed to clean up metadata in the cold tier.
FAILED_INDEX_CREATION - Failed to create an index in the warm tier.
```

Memulihkan indeks dingin dari snapshot

Jika Anda perlu mengembalikan indeks dingin yang dihapus, Anda dapat mengembalikannya kembali ke tingkat hangat dengan mengikuti instruksi the section called “Memulihkan indeks hangat dari

snapshot" dan kemudian memigrasikan indeks kembali ke tingkat dingin lagi. Anda tidak dapat mengembalikan indeks dingin yang dihapus langsung kembali ke tingkat dingin. OpenSearch Layanan mempertahankan indeks dingin selama 14 hari setelah dihapus.

Membatalkan migrasi dari penyimpanan dingin ke hangat

Jika migrasi indeks dari penyimpanan dingin ke penyimpanan hangat diantrekan atau dalam status gagal, Anda dapat membatalkannya dengan permintaan berikut:

```
POST _cold/migration/my-index/_cancel

{
  "acknowledged" : true
}
```

Untuk membatalkan migrasi untuk kumpulan indeks (maksimum 10 pada satu waktu), tentukan ID migrasi:

```
POST _cold/migration/_cancel?migration_id=my-migration-id

{
  "acknowledged" : true
}
```

Mengambil ID migrasi menggunakan permintaan status.

Memperbarui metadata indeks dingin

Anda dapat memperbarui bidang `start_time` dan `end_time` untuk indeks dingin:

```
PATCH _cold/my-index
 {
 "start_time": "2020-01-01",
 "end_time": "2020-02-01"
 }
```

Anda tidak dapat memperbarui `timestamp_field` dari indeks dalam penyimpanan dingin.

> **Note**
>
> OpenSearch Dasbor tidak mendukung metode PATCH. Gunakan curl, Postman, atau beberapa metode lain untuk memperbarui metadata dingin.

Menghapus indeks dingin

Jika Anda tidak menggunakan kebijakan ISM, Anda dapat menghapus indeks dingin secara manual. Permintaan berikut menghapus indeks dingin:

```
DELETE _cold/my-index

{
  "acknowledged" : true
}
```

Menonaktifkan penyimpanan dingin

Konsol OpenSearch Layanan adalah cara paling sederhana untuk menonaktifkan penyimpanan dingin. Pilih domain dan pilih Tindakan, Edit konfigurasi klaster, lalu batalkan pilihan Aktifkan penyimpanan dingin.

Untuk menggunakan CLI AWS atau API konfigurasi, pada `ColdStorageOptions`, atur `"Enabled"="false"`.

Sebelum Anda menonaktifkan penyimpanan dingin, Anda harus menghapus semua indeks dingin atau memigrasikannya kembali ke penyimpanan hangat, jika tidak, tindakan penonaktifan gagal.

Penyimpanan OR1 untuk Layanan Amazon OpenSearch

OR1 adalah keluarga instance untuk Amazon OpenSearch Service yang menyediakan cara hemat biaya untuk menyimpan data dalam jumlah besar. Domain dengan instans OR1 menggunakan Amazon Elastic Block Store (Amazon `gp3` EBS) `io1` atau volume untuk penyimpanan utama, dengan data disalin secara sinkron ke Amazon S3 saat tiba. Struktur penyimpanan ini memberikan peningkatan throughput pengindeksan dengan daya tahan tinggi. Keluarga instans OR1 juga mendukung pemulihan data otomatis jika terjadi kegagalan. Untuk informasi tentang opsi tipe instans OR1, lihatthe section called “Jenis instance generasi saat ini”.

Jika Anda menjalankan pengindeksan beban kerja analitik operasional yang berat seperti analitik log, observabilitas, atau analitik keamanan, Anda bisa mendapatkan keuntungan dari peningkatan kinerja dan efisiensi komputasi instans OR1. Selain itu, pemulihan data otomatis yang ditawarkan oleh instans OR1 meningkatkan keandalan keseluruhan domain Anda.

OpenSearch Layanan mengirimkan metrik OR1 terkait penyimpanan ke Amazon. CloudWatch Untuk daftar metrik yang tersedia, lihat???.

Instans OR1 tersedia sesuai permintaan atau dengan harga Instans Cadangan, dengan tarif per jam untuk instans dan penyimpanan yang disediakan di Amazon EBS dan Amazon S3.

Topik

- Batasan
- Bagaimana OR1 berbeda dari penyimpanan UltraWarm
- Menggunakan instans OR1

Batasan

Pertimbangkan batasan berikut saat menggunakan instans OR1 untuk domain Anda.

- Domain Anda harus menjalankan OpenSearch versi 2.11 atau lebih tinggi.
- Domain Anda harus mengaktifkan enkripsi saat istirahat. Untuk informasi selengkapnya, lihat ???.
- Domain Anda harus menjadi domain baru. Anda tidak dapat mengubah domain yang ada untuk menggunakan instans OR1.
- Jika domain Anda menggunakan node master khusus, mereka harus menggunakan instance Graviton. Untuk informasi selengkapnya tentang node master khusus, lihat???.
- Ukuran pecahan pada instans OR1 harus lebih kecil dari 100 GiB. Pecahan yang lebih besar dari 100 GiB dapat memperlambat waktu pemulihan. Jika Anda membuat pecahan yang lebih besar dari 100 GiB pada instans OR1 OpenSearch , Service memblokir permintaan penulisan ke domain. Jika Anda masih ingin menggunakan pecahan yang lebih besar dari 100 GiB, AWS Supporthubungi untuk meminta kenaikan kuota.
- Interval penyegaran untuk indeks pada instans OR1 harus 10 detik atau lebih tinggi. Interval penyegaran default untuk instans OR1 adalah 10 detik.

Bagaimana OR1 berbeda dari penyimpanan UltraWarm

OpenSearch Layanan menyediakan UltraWarm contoh yang dioptimalkan untuk mengurangi biaya penyimpanan data hangat. Baik OR1 dan UltraWarm instans menyimpan data secara lokal di Amazon EBS dan dari jarak jauh di Amazon S3. Namun, OR1 dan UltraWarm instance berbeda dalam beberapa cara penting:

- Instans OR1 menyimpan salinan data di penyimpanan lokal dan jarak jauh. UltraWarm Misalnya, untuk mengurangi biaya penyimpanan, simpan data terutama di penyimpanan jarak jauh. Tergantung pada pola penggunaan, mereka mungkin memindahkannya ke penyimpanan lokal.
- Instans OR1 aktif dan dapat menerima operasi baca dan tulis, sedangkan data pada UltraWarm instance hanya-baca hingga Anda memindahkannya kembali ke penyimpanan panas secara manual.
- UltraWarm bergantung pada snapshot indeks untuk daya tahan data. Contoh OR1, sebagai perbandingan, melakukan replikasi dan pemulihan di belakang layar. Jika terjadi indeks merah, instans OR1 secara otomatis mengembalikan pecahan yang hilang dari penyimpanan jarak jauh di Amazon S3. Waktu pemulihan bervariasi tergantung pada volume data yang akan dipulihkan.

Untuk informasi selengkapnya tentang UltraWarm penyimpanan, lihat???.

Menggunakan instans OR1

Anda dapat memilih instans OR1 untuk node data saat membuat domain baru dengan, AWS Command Line Interface (AWS CLI) AWS Management Console, atau SDK. AWS Anda kemudian dapat mengindeks dan menanyakan data menggunakan alat yang ada.

Konsol

1. Arahkan ke konsol OpenSearch Layanan Amazon dihttps://console.aws.amazon.com/aos/.
2. Di panel navigasi kiri, pilih Domain.
3. Pilih Create domain (Buat domain).
4. Masukkan nama untuk domain Anda bersama dengan opsi pilihan Anda lainnya. Di bawah keluarga Instance, pilih OR1. Pilih Buat untuk memulai proses pembuatan domain.

AWS CLI

1. Arahkan ke AWS CLI terminal Anda. Jika Anda perlu menginstal AWS CLI, lihat Menginstal atau memperbarui versi terbaru dari file AWS CLI.
2. Untuk menggunakan penyimpanan OR1, Anda harus memberikan nilai ukuran tipe instans OR1 tertentu di `InstanceType` bidang saat Anda membuat domain. Anda juga harus mengaktifkan enkripsi saat istirahat.

   Contoh berikut membuat domain dengan contoh OR1 ukuran. `2xlarge`

```
aws opensearch create-domain \
  --domain-name test-domain \
  --engine-version OpenSearch_2.11 \
  --cluster-config
 "InstanceType=or1.2xlarge.search,InstanceCount=3,DedicatedMasterEnabled=true,DedicatedMast
 \
  --ebs-options "EBSEnabled=true,VolumeType=gp3,VolumeSize=200" \
  --encryption-at-rest-options Enabled=true \
  --advanced-security-options
 "Enabled=true,InternalUserDatabaseEnabled=true,MasterUserOptions={MasterUserName=test-
user,MasterUserPassword=test-password}" \
  --node-to-node-encryption-options Enabled=true \
  --domain-endpoint-options EnforceHTTPS=true \
  --access-policies '{"Version":"2012-10-17","Statement":
[{"Effect":"Allow","Principal":
{"AWS":"*"},"Action":"es:*","Resource":"arn:aws:es:us-east-1:account-
id:domain/test-domain/*"}]}'
```

Manajemen Status Indeks di OpenSearch Layanan Amazon

Index State Management (ISM) di Amazon OpenSearch Service memungkinkan Anda menentukan kebijakan manajemen kustom yang mengotomatiskan tugas rutin, dan menerapkannya pada indeks dan pola indeks. Anda tidak perlu lagi mengatur dan mengelola proses eksternal untuk menjalankan operasi indeks Anda.

Sebuah kebijakan berisi keadaan default dan daftar keadaan untuk indeks untuk transisi diantaranya. Dalam setiap keadaan, Anda dapat menentukan daftar tindakan untuk melakukan dan kondisi yang memicu transisi ini. Kasus penggunaan yang khas adalah menghapus indeks lama secara berkala

setelah jangka waktu tertentu. Misalnya, Anda dapat menentukan kebijakan yang memindahkan indeks Anda ke keadaan `read_only` setelah 30 hari dan akhirnya menghapusnya setelah 90 hari.

Setelah Anda melampirkan kebijakan ke indeks, ISM membuat pekerjaan yang berjalan setiap 5 hingga 8 menit (atau 30 hingga 48 menit untuk kluster pra-1,3) untuk melakukan tindakan kebijakan, memeriksa kondisi, dan mengalihkan indeks ke negara bagian yang berbeda. Waktu dasar untuk pekerjaan ini dijalankan adalah setiap 5 menit, ditambah jitter acak 0-60% ditambahkan ke dalamnya untuk memastikan Anda tidak melihat lonjakan aktivitas dari semua indeks Anda pada saat yang bersamaan. ISM tidak menjalankan pekerjaan jika keadaan klaster berwarna merah.

ISM membutuhkan OpenSearch atau Elasticsearch 6.8 atau yang lebih baru. Dokumentasi lengkap tersedia dalam OpenSearchdokumentasi.

> **Important**
>
> Anda tidak dapat lagi menggunakan templat indeks untuk menerapkan kebijakan ISM ke indeks yang baru dibuat. Anda dapat terus mengelola indeks yang baru dibuat secara otomatis dengan bidang template ISM. Pemutakhiran ini memperkenalkan perubahan melanggar yang mempengaruhi CloudFormation template yang ada menggunakan pengaturan ini.

Membuat kebijakan ISM

Untuk memulai dengan Manajemen Negara Indeks

1. Buka konsol OpenSearch Layanan Amazon di https://console.aws.amazon.com/aos/home.
2. Pilih domain yang ingin Anda buat kebijakan ISM.
3. Dari dasbor domain, buka URL OpenSearch Dasbor dan masuk dengan nama pengguna dan kata sandi utama Anda. URL mengikuti format ini:

   ```
   domain-endpoint/_dashboards/
   ```

4. Buka panel navigasi kiri dalam OpenSearch Dasbor dan pilih Manajemen Indeks, lalu Buat kebijakan.
5. Gunakan editor visual atau editor JSON untuk membuat kebijakan. Sebaiknya gunakan editor visual karena menawarkan cara yang lebih terstruktur untuk mendefinisikan kebijakan. Untuk bantuan membuat kebijakan, lihat contoh kebijakan di bawah ini.

6. Setelah Anda membuat kebijakan, lampirkan ke satu atau lebih indeks:

```
POST _plugins/_ism/add/my-index
{
  "policy_id": "my-policy-id"
}
```

> **Note**
>
> Jika domain Anda menjalankan versi lama Elasticsearch, gunakan `_opendistro` sebagai gantinya. `_plugins`

Atau, pilih indeks di OpenSearch Dasbor dan pilih Terapkan kebijakan.

Contoh kebijakan

Contoh kebijakan berikut menunjukkan caranya mengotomatisasi kasus penggunaan ISM yang umum.

Penyimpanan panas ke hangat ke dingin

Kebijakan sampel ini memindahkan indeks dari penyimpanan panas ke UltraWarm, dan akhirnya penyimpanan dingin. Kemudian, menghapus indeks.

Indeks pada mulanya dalam keadaan `hot`. Setelah sepuluh hari, ISM memindahkannya ke `warm` negara. 80 hari kemudian, setelah indeks berusia 90 hari, ISM memindahkan indeks ke `cold`negara. Setelah satu tahun, layanan tersebut mengirimkan notifikasi ke ruang Amazon Chime bahwa indeks tersebut sedang dihapus dan menghapusnya secara permanen.

Perhatikan bahwa indeks dingin memerlukan `cold_delete` operasi daripada `delete` operasi normal. Perhatikan juga bahwa eksplisit `timestamp_field` diperlukan dalam data Anda untuk mengelola indeks dingin dengan ISM.

```
{
  "policy": {
    "description": "Demonstrate a hot-warm-cold-delete workflow.",
    "default_state": "hot",
    "schema_version": 1,
    "states": [{
```

```
        "name": "hot",
        "actions": [],
        "transitions": [{
          "state_name": "warm",
          "conditions": {
            "min_index_age": "10d"
          }
        }]
      },
      {
        "name": "warm",
        "actions": [{
          "warm_migration": {},
          "retry": {
            "count": 5,
            "delay": "1h"
          }
        }],
        "transitions": [{
          "state_name": "cold",
          "conditions": {
            "min_index_age": "90d"
          }
        }]
      },
      {
        "name": "cold",
        "actions": [{
            "cold_migration": {
              "timestamp_field": "<your timestamp field>"
            }
          }
        ],
        "transitions": [{
          "state_name": "delete",
          "conditions": {
             "min_index_age": "365d"
          }
        }]
      },
      {
        "name": "delete",
        "actions": [{
          "notification": {
```

```
            "destination": {
              "chime": {
                "url": "<URL>"
              }
            },
            "message_template": {
              "source": "The index {{ctx.index}} is being deleted."
            }
          }
        },
        {
          "cold_delete": {}
        }]
      }
    ]
  }
}
```

Kurangi jumlah replika

Kebijakan contoh ini mengurangi jumlah replika ke nol setelah tujuh hari untuk menghemat ruang disk dan menghapus indeks setelah 21 hari. Kebijakan ini berasumsi bahwa indeks Anda tidak kritis dan tidak lagi menerima permintaan tulis; memiliki nol replika mendatangkan risiko kehilangan data.

```
{
  "policy": {
    "description": "Changes replica count and deletes.",
    "schema_version": 1,
    "default_state": "current",
    "states": [{
        "name": "current",
        "actions": [],
        "transitions": [{
          "state_name": "old",
          "conditions": {
            "min_index_age": "7d"
          }
        }]
      },
      {
        "name": "old",
        "actions": [{
          "replica_count": {
```

```
            "number_of_replicas": 0
          }
        }],
        "transitions": [{
          "state_name": "delete",
          "conditions": {
            "min_index_age": "21d"
          }
        }]
      },
      {
        "name": "delete",
        "actions": [{
          "delete": {}
        }],
        "transitions": []
      }
    ]
  }
}
```

Mengambil snapshot indeks

Kebijakan contoh ini menggunakan operasi `snapshot` untuk mengambil snapshot dari indeks segera setelah diisi dengan setidaknya satu dokumen. `repository` adalah nama repositori snapshot manual yang Anda daftarkan di Amazon S3. `snapshot` adalah nama dari snapshot. Untuk prasyarat snapshot dan langkah-langkah untuk mendaftarkan repositori, lihat the section called “Membuat snapshot indeks”.

```
{
  "policy": {
    "description": "Takes an index snapshot.",
    "schema_version": 1,
    "default_state": "empty",
    "states": [{
        "name": "empty",
        "actions": [],
        "transitions": [{
          "state_name": "occupied",
          "conditions": {
            "min_doc_count": 1
          }
        }]
```

```
      },
      {
        "name": "occupied",
        "actions": [{
          "snapshot": {
            "repository": "<my-repository>",
            "snapshot": "<my-snapshot>"
            }
          }],
          "transitions": []
      }
    ]
  }
}
```

Templat ISM

Anda dapat mengatur bidang `ism_template` dalam kebijakan sehingga ketika Anda membuat indeks yang cocok dengan pola templat, kebijakan secara otomatis dilampirkan ke indeks tersebut. Dalam contoh ini, setiap indeks yang Anda buat dengan nama yang dimulai dengan "log" secara otomatis cocok dengan kebijakan ISM `my-policy-id`:

```
PUT _plugins/_ism/policies/my-policy-id
{
  "policy": {
    "description": "Example policy.",
    "default_state": "...",
    "states": [...],
    "ism_template": {
      "index_patterns": ["log*"],
      "priority": 100
    }
  }
}
```

Untuk contoh yang lebih rinci, lihat Contoh kebijakan dengan template ISM untuk rollover otomatis.

Perbedaan

Dibandingkan dengan OpenSearch dan Elasticsearch, ISM for Amazon OpenSearch Service memiliki beberapa perbedaan.

Operasi ISM

- OpenSearchLayanan mendukung tiga operasi ISM yang unik`warm_migration`,,`cold_migration`, dan`cold_delete`:
  - Jika domain Anda telah UltraWarmdiaktifkan, `warm_migration` tindakan akan mengalihkan indeks ke penyimpanan hangat.
  - Jika domain Anda mengaktifkan cold storage, `cold_migration` tindakan akan mengalihkan indeks ke cold storage, dan `cold_delete` tindakan akan menghapus indeks dari cold storage.

  Bahkan jika salah satu tindakan ini tidak selesai dalam periode batas waktu yang ditetapkan, migrasi atau penghapusan indeks masih berlanjut. Menetapkan error_notification untuk salah satu tindakan di atas akan memberi tahu Anda bahwa tindakan gagal jika tidak selesai dalam periode batas waktu, tetapi notifikasi hanya untuk referensi Anda sendiri. Operasi sebenarnya tidak memiliki batas waktu yang melekat dan terus berjalan sampai akhirnya berhasil atau gagal.
- Jika domain Anda berjalan OpenSearch atau Elasticsearch 7.4 atau yang lebih baru, OpenSearch Service mendukung ISM `open` dan operasi. `close`
- Jika domain Anda berjalan OpenSearch atau Elasticsearch 7.7 atau yang lebih baru, OpenSearch Service mendukung operasi ISM. `snapshot`

Operasi ISM penyimpanan dingin

Untuk indeks dingin, Anda harus menentukan `?type=_cold` parameter saat menggunakan API ISM berikut:

- Tambahkan kebijakan
- Hapus kebijakan
- Kebijakan pemutakhiran
- Coba lagi indeks gagal
- Jelaskan indeks

API untuk indeks dingin ini memiliki perbedaan tambahan berikut:

- Operator wildcard tidak didukung kecuali saat Anda menggunakannya di bagian akhir. Misalnya, `_plugins/_ism/<add, remove, change_policy, retry, explain>/logstash-*`

didukung tetapi `_plugins/_ism/<add, remove, change_policy, retry, explain>/iad-*-prod` tidak didukung.

- Beberapa nama indeks dan pola tidak didukung. Misalnya, `_plugins/_ism/<add, remove, change_policy, retry, explain>/app-logs` didukung tetapi `_plugins/_ism/<add, remove, change_policy, retry, explain>/app-logs,sample-data` tidak didukung.

Pengaturan ISM

OpenSearchdan Elasticsearch memungkinkan Anda mengubah semua pengaturan ISM yang tersedia menggunakan API. `_cluster/settings` Di Amazon OpenSearch Service, Anda hanya dapat mengubah pengaturan ISM berikut:

- Pengaturan tingkat kluster:
  - `plugins.index_state_management.enabled`
  - `plugins.index_state_management.history.enabled`
- Pengaturan tingkat indeks:
  - `plugins.index_state_management.rollover_alias`

Tutorial: Mengotomatisasi proses Manajemen Negara Indeks

Tutorial ini menunjukkan bagaimana menerapkan kebijakan ISM yang mengotomatiskan tugas manajemen indeks rutin dan menerapkannya pada indeks dan pola indeks.

Index State Management (ISM) di Amazon OpenSearch Service memungkinkan Anda mengotomatiskan aktivitas manajemen indeks berulang, sehingga Anda dapat menghindari penggunaan alat tambahan untuk mengelola siklus hidup indeks. Anda dapat membuat kebijakan yang mengotomatiskan operasi ini berdasarkan usia indeks, ukuran, dan kondisi lainnya, semuanya dari dalam domain OpenSearch Layanan Amazon Anda.

OpenSearchLayanan mendukung tiga tingkatan penyimpanan: status “panas” default untuk penulisan aktif dan analisis latensi rendah, UltraWarm untuk data hanya-baca hingga tiga petabyte, dan penyimpanan dingin untuk arsip jangka panjang tanpa batas.

Tutorial ini menyajikan contoh kasus penggunaan penanganan data waktu-seri dalam indeks harian. Dalam tutorial ini, Anda menyiapkan kebijakan yang mengambil snapshot otomatis dari setiap indeks terlampir setelah 24 jam. Kemudian bermigrasi indeks dari keadaan panas default ke UltraWarm

penyimpanan setelah dua hari, cold storage setelah 30 hari, dan akhirnya menghapus indeks setelah 60 hari.

Prasyarat

- Domain OpenSearch Layanan Anda harus menjalankan Elasticsearch versi 6.8 atau yang lebih baru.
- Domain Anda harus mengaktifkan cold storage UltraWarmdan cold storage.
- Anda harus mendaftarkan repositori snapshot manual untuk domain Anda.
- Peran pengguna Anda memerlukan izin yang cukup untuk mengakses konsol OpenSearch Layanan. Jika perlu, validasi dan konfigurasikan akses ke domain Anda.

Langkah 1: Mengkonfigurasi kebijakan ISM

Pertama, konfigurasikan kebijakan ISM di OpenSearch Dasbor.

1. Dari dasbor domain Anda di konsol OpenSearch Layanan, buka URL OpenSearch Dasbor dan masuk dengan nama pengguna dan kata sandi utama Anda. URL mengikuti format ini:*domain-endpoint*/_dashboards/.
2. Di OpenSearch Dasbor, pilih Tambahkan data sampel dan tambahkan satu atau beberapa indeks sampel ke domain Anda.
3. Buka panel navigasi kiri dan pilih Manajemen Indeks, lalu pilih Buat kebijakan.
4. Sebutkan kebijakan `ism-policy-example`.
5. Ganti kebijakan default dengan kebijakan berikut:

```
{
  "policy": {
    "description": "Move indexes between storage tiers",
    "default_state": "hot",
    "states": [
      {
        "name": "hot",
        "actions": [],
        "transitions": [
          {
            "state_name": "snapshot",
            "conditions": {
              "min_index_age": "24h"
```

```
            }
          }
        ]
      },
      {
        "name": "snapshot",
        "actions": [
          {
            "retry": {
              "count": 5,
              "backoff": "exponential",
              "delay": "30m"
            },
            "snapshot": {
              "repository": "snapshot-repo",
              "snapshot": "ism-snapshot"
            }
          }
        ],
        "transitions": [
          {
            "state_name": "warm",
            "conditions": {
              "min_index_age": "2d"
            }
          }
        ]
      },
      {
        "name": "warm",
        "actions": [
          {
            "retry": {
              "count": 5,
              "backoff": "exponential",
              "delay": "1h"
            },
            "warm_migration": {}
          }
        ],
        "transitions": [
          {
            "state_name": "cold",
            "conditions": {
```

```
            "min_index_age": "30d"
          }
        }
      ]
    },
    {
      "name": "cold",
      "actions": [
        {
          "retry": {
            "count": 5,
            "backoff": "exponential",
            "delay": "1h"
          },
          "cold_migration": {
            "start_time": null,
            "end_time": null,
            "timestamp_field": "@timestamp",
            "ignore": "none"
          }
        }
      ],
      "transitions": [
        {
          "state_name": "delete",
          "conditions": {
            "min_index_age": "60d"
          }
        }
      ]
    },
    {
      "name": "delete",
      "actions": [
        {
          "cold_delete": {}
        }
      ],
      "transitions": []
    }
  ],
  "ism_template": [
    {
      "index_patterns": [
```

```
          "index-*"
        ],
        "priority": 100
      }
    ]
  }
}
```

> **Note**
>
> `ism_template`Bidang secara otomatis melampirkan kebijakan ke indeks yang baru dibuat yang cocok dengan salah satu yang ditentukan`index_patterns`. Dalam hal ini, semua indeks yang dimulai dengan`index-`. Anda dapat memodifikasi bidang ini agar sesuai dengan format indeks di lingkungan Anda. Untuk informasi selengkapnya, lihat template ISM.

6. Di `snapshot` bagian kebijakan, ganti *`snapshot-repo`* dengan nama repositori snapshot yang Anda daftarkan untuk domain Anda. Anda juga dapat mengganti secara opsional*`ism-snapshot`*, yang akan menjadi nama snapshot saat dibuat.
7. Pilih Create (Buat). Kebijakan ini sekarang terlihat di halaman kebijakan manajemen negara.

Langkah 2: Lampirkan kebijakan ke satu atau lebih indeks

Sekarang setelah Anda membuat kebijakan Anda, lampirkan ke satu atau lebih indeks di klaster Anda.

1. Buka tab Hot indicies dan cari`opensearch_dashboards_sample`, yang mencantumkan semua indeks sampel yang Anda tambahkan pada langkah 1.
2. Pilih semua indeks dan pilih Terapkan kebijakan, lalu pilih ism-policy-examplekebijakan yang baru saja Anda buat.
3. Pilih Apply (Terapkan).

Anda dapat memantau indeks saat mereka bergerak melalui berbagai status pada halaman Indeks yang dikelola kebijakan.

Meringkas indeks di Amazon OpenSearch Service dengan rollups indeks

Indeks rollups di Amazon OpenSearch Service memungkinkan Anda mengurangi biaya penyimpanan dengan secara berkala menggulung data lama menjadi indeks yang diringkas.

Anda memilih bidang yang menarik minat Anda dan menggunakan indeks rollup untuk membuat indeks baru dengan hanya bidang-bidang yang dikumpulkan ke dalam bucket waktu kasar. Anda dapat menyimpan bulan atau tahun data historis di sebagian kecil dari biaya dengan performa kueri yang sama.

Indeks rollups membutuhkan OpenSearch atau Elasticsearch 7.9 atau yang lebih baru. Dokumentasi lengkap pada fitur tersedia di OpenSearchdokumentasi.

Membuat pekerjaan indeks rollup

Untuk memulai, pilih Manajemen Indeks di OpenSearch Dasbor. Pilih Pekerjaan Rollup dan pilih Buat pekerjaan rollup.

Langkah 1: Siapkan indeks

Mengatur indeks sumber dan target. Indeks sumber adalah salah satu yang ingin Anda gulung. Indeks target adalah di mana hasil penggulungan indeks disimpan.

Setelah Anda membuat pekerjaan indeks rollup, Anda tidak dapat mengubah pilihan indeks Anda.

Langkah 2: Tentukan agregasi dan metrik

Pilih atribut dengan agregasi (istilah dan histogram) dan metrik (rata-rata, jumlah, maks, menit, dan jumlah nilai) yang ingin Anda gulung. Pastikan Anda tidak menambahkan banyak atribut yang sangat rinci, karena Anda tidak akan menghemat banyak ruang.

Langkah 3: Tentukan jadwal

Tentukan jadwal untuk menggulung indeks Anda saat sedang tertelan. Pekerjaan indeks rollup diaktifkan secara default.

Langkah 4: Tinjau dan buat

Tinjau konfigurasi Anda dan pilih Buat.

Langkah 5: Cari indeks target

Anda bisa menggunakan API `_search` standar untuk mencari indeks target. Anda tidak dapat mengakses struktur internal data dalam indeks target karena plugin secara otomatis menulis ulang kueri di latar belakang agar sesuai dengan indeks target. Hal ini untuk memastikan Anda dapat menggunakan kueri yang sama untuk sumber dan target indeks.

Untuk kueri indeks target, atur `size` ke 0:

```
GET target_index/_search
{
  "size": 0,
  "query": {
    "match_all": {}
  },
  "aggs": {
    "avg_cpu": {
      "avg": {
        "field": "cpu_usage"
      }
    }
  }
}
```

> **Note**
>
> OpenSearch versi 2.2 dan yang lebih baru mendukung pencarian beberapa indeks rollup dalam satu permintaan. OpenSearch versi sebelum 2.2 dan versi OSS Elasticsearch lama hanya mendukung satu indeks rollup per pencarian.

Mengubah indeks di AmazonOpenSearchLayanan

Padahalpekerjaan indeks rollupmemungkinkan Anda mengurangi granularitas data dengan menggulung data lama ke dalam indeks kental, mengubah pekerjaan memungkinkan Anda membuat tampilan data yang berbeda dan diringkas yang berpusat di sekitar bidang tertentu, sehingga Anda dapat memvisualisasikan atau menganalisis data dengan cara yang berbeda.

Transformasi indeks memilikiOpenSearchDashboard antarmuka pengguna dan REST API. Fitur ini membutuhkanOpenSearch1.0 atau yang lebih baru. Dokumentasi lengkap tersedia diOpenSearchdokumentasi.

Membuat pekerjaan indeks

Jika Anda tidak memiliki data apa pun di klaster Anda, gunakan sampel data penerbangan di dalamnyaOpenSearchDasbor untuk mencoba mengubah pekerjaan. Setelah menambahkan data, luncurkanOpenSearchDasbor. Kemudian pilihManajemen Indeks,Pekerjaan Transformasi, danMembuat Job Transformasi.

Langkah 1: Pilih indeks

DiIndeksbagian, pilih indeks sumber dan target. Anda dapat memilih indeks target yang sudah ada atau membuat indeks target baru dengan memasukkan nama untuk indeks target yang sudah ada.

Jika Anda ingin mengubah hanya subset dari indeks sumber Anda, pilihMenambahkan Filter Data, dan menggunakanOpenSearch kueri DSLuntuk menentukan subset dari indeks sumber Anda.

Langkah 2: Pilih bidang

Setelah memilih indeks Anda, pilih bidang yang ingin Anda gunakan dalam pekerjaan transformasi Anda, serta apakah akan menggunakan pengelompokan atau agregasi.

- Anda dapat menggunakan pengelompokan untuk menempatkan data Anda ke dalam ember terpisah dalam indeks yang ditransformasikan. Misalnya, jika Anda ingin mengelompokkan semua tujuan bandara dalam data penerbangan sampel, kelompokkan`DestAirportID`bidang ke bidang target`DestAirportID_terms`lapangan, dan Anda dapat menemukan ID bandara yang dikelompokkan dalam indeks Anda berubah setelah pekerjaan transformasi selesai.
- Di sisi lain, agregasi memungkinkan Anda melakukan perhitungan sederhana. Misalnya, Anda mungkin menyertakan agregasi dalam pekerjaan transformasi Anda untuk menentukan bidang baru`sum_of_total_ticket_price`yang menghitung jumlah semua tiket pesawat. Kemudian Anda dapat menganalisis data baru dalam indeks Anda yang ditransformasikan.

Langkah 3: Tentukan jadwal

Transform pekerjaan diaktifkan secara default dan berjalan pada jadwal. UntukTransformasi interval eksekusi, tentukan interval dalam menit, jam, atau hari.

Langkah 4: Memeriksa dan memantau

Tinjau konfigurasi Anda dan pilih Buat. Kemudian pantauMengubah status tugaskolom.

Langkah 5: Cari indeks target

Setelah pekerjaan selesai, Anda dapat menggunakan standar`_search`API untuk mencari indeks target.

Misalnya, setelah menjalankan pekerjaan transformasi yang mengubah data penerbangan berdasarkan`DestAirportID`lapangan, Anda dapat menjalankan permintaan berikut untuk mengembalikan semua bidang yang memiliki nilai`SFO`:

```
GET target_index/_search
{
  "query": {
    "match": {
      "DestAirportID_terms" : "SFO"
    }
  }
}
```

Replikasi lintas cluster untuk Layanan Amazon OpenSearch

Dengan replikasi lintas klaster di Amazon OpenSearch Service, Anda dapat mereplikasi indeks, pemetaan, dan metadata pengguna dari satu domain Layanan ke domain Layanan lainnya. OpenSearch Menggunakan replikasi lintas cluster membantu memastikan pemulihan bencana jika terjadi pemadaman, dan memungkinkan Anda mereplikasi data di seluruh pusat data yang jauh secara geografis untuk mengurangi latensi. Anda membayar biaya transfer AWS data standar untuk data yang ditransfer antar domain.

Replikasi lintas cluster mengikuti model replikasi aktif-pasif di mana indeks lokal atau pengikut menarik data dari indeks jarak jauh atau pemimpin. Indeks pemimpin mengacu pada sumber data, atau indeks tempat Anda ingin mereplikasi data. Indeks pengikut mengacu pada target untuk data, atau indeks yang ingin Anda replikasi data.

Replikasi lintas-cluster tersedia di domain yang menjalankan Elasticsearch 7.10 atau 1.1 atau yang lebih baru. OpenSearch Dokumentasi lengkap untuk replikasi lintas cluster tersedia dalam dokumentasi. OpenSearch

Topik

Batasan

Replikasi lintas cluster memiliki keterbatasan sebagai berikut:

- Anda tidak dapat mereplikasi data antara domain OpenSearch Layanan Amazon dan cluster yang dikelola sendiri OpenSearch atau Elasticsearch.
- Anda tidak dapat mereplikasi indeks dari domain pengikut ke domain pengikut lain. Jika Anda ingin mereplikasi indeks ke beberapa domain pengikut, Anda hanya dapat mereplikasi indeks dari domain pemimpin tunggal.
- Sebuah domain dapat dihubungkan, melalui kombinasi koneksi inbound dan outbound, hingga maksimal 20 domain lainnya.
- Saat Anda pertama kali menyiapkan koneksi lintas-cluster, domain pemimpin harus berada pada versi yang sama atau lebih tinggi dari domain pengikut.
- Anda tidak dapat menggunakan AWS CloudFormation untuk menghubungkan domain.
- Anda tidak dapat menggunakan replikasi lintas cluster pada instance M3 atau burstable (T2 dan T3).
- Anda tidak dapat mereplikasi data antara UltraWarm atau indeks dingin. Kedua indeks harus dalam penyimpanan panas.
- Saat Anda menghapus indeks dari domain pemimpin, indeks yang sesuai pada domain pengikut tidak akan dihapus secara otomatis.

Prasyarat

Sebelum menyiapkan replikasi lintas klaster, pastikan domain Anda memenuhi persyaratan berikut:

- Elasticsearch 7.10 atau 1.1 atau OpenSearch yang lebih baru
- Kontrol akses berbutir halus diaktifkan
- ode-to-node Enkripsi N diaktifkan

Persyaratan izin

Untuk memulai replikasi, Anda harus menyertakan `es:ESCrossClusterGet` izin pada domain jarak jauh (pemimpin). Kami merekomendasikan kebijakan IAM berikut pada domain jarak jauh. Kebijakan ini juga memungkinkan Anda melakukan operasi lain, seperti mengindeks dokumen dan melakukan penelusuran standar:

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Effect": "Allow",
      "Principal": {
        "AWS": [
          "*"
        ]
      },
      "Action": [
        "es:ESHttp*"
      ],
      "Resource": "arn:aws:es:region:account:domain/leader-domain/*"
    },
    {
      "Effect": "Allow",
      "Principal": {
        "AWS": "*"
      },
      "Action": "es:ESCrossClusterGet",
      "Resource": "arn:aws:es:region:account:domain/leader-domain"
    }
  ]
}
```

Pastikan bahwa izin `es:ESCrossClusterGet` diterapkan untuk `/leader-domain` dan bukan `/leader-domain/*`.

Agar pengguna non-admin dapat melakukan aktivitas replikasi, mereka juga perlu dipetakan ke izin yang sesuai. Sebagian besar izin sesuai dengan operasi REST API tertentu. Misalnya, `indices:admin/plugins/replication/index/_resume` izin memungkinkan Anda melanjutkan replikasi indeks. Untuk daftar lengkap izin, lihat Izin replikasi dalam dokumentasi. OpenSearch

> **Note**
>
> Perintah untuk memulai replikasi dan membuat aturan replikasi adalah kasus khusus. Karena mereka memanggil proses latar belakang pada domain pemimpin dan pengikut, Anda harus lulus `leader_cluster_role` dan `follower_cluster_role` dalam permintaan. OpenSearch Layanan menggunakan peran ini dalam semua tugas replikasi backend. Untuk informasi tentang pemetaan dan penggunaan peran ini, lihat Memetakan peran klaster pemimpin dan pengikut dalam dokumentasi. OpenSearch

Siapkan koneksi lintas-cluster

Untuk mereplikasi indeks dari satu domain ke domain lainnya, Anda perlu menyiapkan koneksi lintas-cluster antar domain. Cara termudah untuk menghubungkan domain adalah melalui tab Koneksi pada dasbor domain. Anda juga dapat menggunakan API konfigurasi atau AWSCLI. Karena replikasi lintas cluster mengikuti model “tarik”, Anda memasukkan koneksi dari domain pengikut.

> **Note**
>
> Jika sebelumnya Anda menghubungkan dua domain untuk melakukan pencarian lintas klaster, Anda tidak dapat menggunakan koneksi yang sama untuk replikasi. Koneksi ditandai seperti `SEARCH_ONLY` di konsol. Untuk melakukan replikasi antara dua domain yang terhubung sebelumnya, Anda harus menghapus koneksi dan membuatnya kembali. Setelah Anda melakukan ini, koneksi tersedia untuk pencarian lintas cluster dan replikasi lintas-cluster.

Untuk mengatur koneksi

1. Di konsol OpenSearch Layanan Amazon, pilih domain pengikut, buka tab Koneksi, dan pilih Permintaan.
2. Untuk alias Koneksi, masukkan nama untuk koneksi Anda.
3. Pilih antara menghubungkan ke domain di wilayah Anda Akun AWS atau di akun atau Wilayah lain.

   - Untuk terhubung ke domain di Wilayah Akun AWS dan Anda, pilih domain dan pilih Permintaan.
   - Untuk terhubung ke domain di wilayah lain Akun AWS atau wilayah, tentukan ARN dari domain jarak jauh dan pilih Permintaan.

OpenSearch Layanan memvalidasi permintaan koneksi. Jika domain tidak kompatibel, koneksi gagal. Jika validasi berhasil, validasi dikirim ke domain tujuan untuk persetujuan. Ketika domain tujuan menyetujui permintaan, Anda dapat memulai replikasi.

Replikasi lintas cluster mendukung replikasi dua arah. Ini berarti Anda dapat membuat koneksi keluar dari domain A ke domain B, dan koneksi keluar lainnya dari domain B ke domain A. Anda kemudian dapat mengatur replikasi sehingga domain A mengikuti indeks di domain B, dan domain B mengikuti indeks di domain A.

Memulai replikasi

Setelah Anda membuat koneksi lintas cluster, Anda dapat mulai mereplikasi data. Pertama, buat indeks pada domain pemimpin untuk ditiru:

```
PUT leader-01
```

Untuk mereplikasi indeks itu, kirim perintah ini ke domain pengikut:

```
PUT _plugins/_replication/follower-01/_start
{
   "leader_alias": "connection-alias",
   "leader_index": "leader-01",
   "use_roles":{
      "leader_cluster_role": "all_access",
      "follower_cluster_role": "all_access"
   }
```

```
}
```

Anda dapat menemukan alias koneksi di tab Connections di dasbor domain Anda.

Contoh ini mengasumsikan bahwa admin mengeluarkan permintaan dan penggunaan `all_access` untuk dan untuk `leader_cluster_role` kesederhanaan`follower_cluster_role`. Namun, di lingkungan produksi, kami menyarankan Anda membuat pengguna replikasi pada indeks pemimpin dan pengikut, dan memetakannya sesuai dengan itu. Nama pengguna harus identik. Untuk informasi tentang peran ini dan cara memetakannya, lihat Memetakan peran klaster pemimpin dan pengikut dalam OpenSearch dokumentasi.

Konfirmasikan replikasi

Untuk mengonfirmasi bahwa replikasi sedang terjadi, dapatkan status replikasi:

```
GET _plugins/_replication/follower-01/_status

{
  "status" : "SYNCING",
  "reason" : "User initiated",
  "leader_alias" : "connection-alias",
  "leader_index" : "leader-01",
  "follower_index" : "follower-01",
  "syncing_details" : {
    "leader_checkpoint" : -5,
    "follower_checkpoint" : -5,
    "seq_no" : 0
  }
}
```

Nilai pos pemeriksaan pemimpin dan pengikut dimulai sebagai bilangan bulat negatif dan mencerminkan jumlah pecahan yang Anda miliki (-1 untuk satu pecahan, -5 untuk lima pecahan, dan seterusnya). Nilai bertambah menjadi bilangan bulat positif dengan setiap perubahan yang Anda buat. Jika nilainya sama, itu berarti indeks disinkronkan sepenuhnya. Anda dapat menggunakan nilai pos pemeriksaan ini untuk mengukur latensi replikasi di seluruh domain Anda.

Untuk memvalidasi replikasi lebih lanjut, tambahkan dokumen ke indeks pemimpin:

```
PUT leader-01/_doc/1
{
   "Doctor Sleep":"Stephen King"
```

```
}
```

Dan konfirmasikan bahwa itu muncul di indeks pengikut:

```
GET follower-01/_search

{
    ...
    "max_score" : 1.0,
    "hits" : [
      {
        "_index" : "follower-01",
        "_type" : "_doc",
        "_id" : "1",
        "_score" : 1.0,
        "_source" : {
          "Doctor Sleep" : "Stephen King"
        }
      }
    ]
  }
}
```

Jeda dan lanjutkan replikasi

Anda dapat menghentikan sementara replikasi jika perlu memperbaiki masalah atau mengurangi beban pada domain pemimpin. Kirim permintaan ini ke domain pengikut. Pastikan untuk menyertakan badan permintaan kosong:

```
POST _plugins/_replication/follower-01/_pause
{}
```

Kemudian dapatkan status untuk memastikan bahwa replikasi dijeda:

```
GET _plugins/_replication/follower-01/_status

{
  "status" : "PAUSED",
  "reason" : "User initiated",
  "leader_alias" : "connection-alias",
  "leader_index" : "leader-01",
  "follower_index" : "follower-01"
```

```
}
```

Setelah selesai membuat perubahan, lanjutkan replikasi. Kirim permintaan ini ke domain pengikut. Pastikan untuk menyertakan badan permintaan kosong:

```
POST _plugins/_replication/follower-01/_resume
{}
```

Anda tidak dapat melanjutkan replikasi setelah dijeda selama lebih dari 12 jam. Anda harus menghentikan replikasi, menghapus indeks pengikut, dan memulai ulang replikasi pemimpin.

Hentikan replikasi

Ketika Anda menghentikan replikasi sepenuhnya, indeks pengikut berhenti mengikuti pemimpin dan menjadi indeks standar. Anda tidak dapat memulai ulang replikasi setelah Anda menghentikannya.

Hentikan replikasi dari domain pengikut. Pastikan untuk menyertakan badan permintaan kosong:

```
POST _plugins/_replication/follower-01/_stop
{}
```

Ikuti otomatis

Anda dapat menentukan seperangkat aturan replikasi terhadap domain pemimpin tunggal yang secara otomatis mereplikasi indeks yang cocok dengan pola tertentu. Ketika indeks pada domain pemimpin cocok dengan salah satu pola (misalnya,`books*`), indeks pengikut yang cocok dibuat pada domain pengikut. OpenSearch Layanan mereplikasi indeks yang ada yang cocok dengan pola, serta indeks baru yang Anda buat. Itu tidak mereplikasi indeks yang sudah ada di domain pengikut.

Untuk mereplikasi semua indeks (dengan pengecualian indeks yang dibuat sistem, dan indeks yang sudah ada di domain pengikut), gunakan pola wildcard (). *

Buat aturan replikasi

Buat aturan replikasi pada domain pengikut, dan tentukan nama koneksi lintas-cluster:

```
POST _plugins/_replication/_autofollow
{
   "leader_alias" : "connection-alias",
   "name": "rule-name",
```

```
   "pattern": "books*",
   "use_roles":{
      "leader_cluster_role": "all_access",
      "follower_cluster_role": "all_access"
   }
}
```

Anda dapat menemukan alias koneksi di tab Connections di dasbor domain Anda.

Contoh ini mengasumsikan bahwa admin mengeluarkan permintaan, dan digunakan `all_access` sebagai peran domain pemimpin dan pengikut untuk kesederhanaan. Namun, di lingkungan produksi, kami menyarankan Anda membuat pengguna replikasi pada indeks pemimpin dan pengikut dan memetakannya sesuai dengan itu. Nama pengguna harus identik. Untuk informasi tentang peran ini dan cara memetakannya, lihat Memetakan peran klaster pemimpin dan pengikut dalam OpenSearch dokumentasi.

Untuk mengambil daftar aturan replikasi yang ada di domain, gunakan operasi API statistik ikuti otomatis.

Untuk menguji aturan, buat indeks yang cocok dengan pola pada domain pemimpin:

```
PUT books-are-fun
```

Dan periksa apakah replika muncul di domain pengikut:

```
GET _cat/indices

health status index          uuid                   pri rep docs.count docs.deleted
 store.size pri.store.size
green  open   books-are-fun  ldfHO78xYYdxRMULuiTvSQ     1   1          0            0
     208b           208b
```

Hapus aturan replikasi

Saat Anda menghapus aturan replikasi, OpenSearch Service berhenti mereplikasi indeks baru yang cocok dengan pola, tetapi melanjutkan aktivitas replikasi yang ada hingga Anda menghentikan replikasi indeks tersebut.

Hapus aturan replikasi dari domain pengikut:

```
DELETE _plugins/_replication/_autofollow
```

```
{
   "leader_alias" : "connection-alias",
   "name": "rule-name"
}
```

Meningkatkan domain yang terhubung

Untuk meningkatkan versi mesin dari dua domain yang memiliki koneksi lintas-cluster, tingkatkan domain pengikut terlebih dahulu dan kemudian domain pemimpin. Jangan hapus koneksi di antara mereka, jika tidak replikasi berhenti dan Anda tidak akan dapat melanjutkannya.

Memigrasi indeks OpenSearch Layanan Amazon menggunakan indeks ulang jarak jauh

Remote reindex memungkinkan Anda menyalin indeks dari satu domain OpenSearch Layanan Amazon ke domain lain. Anda dapat memigrasikan indeks dari domain OpenSearch Layanan atau cluster yang dikelola sendiri OpenSearch dan Elasticsearch.

Domain dan indeks jarak jauh mengacu pada sumber data, atau domain dan indeks tempat Anda ingin menyalin data. Domain dan indeks lokal mengacu pada target untuk data, atau domain dan indeks yang ingin Anda salin datanya.

Pengindeksan ulang jarak jauh membutuhkan OpenSearch 1.0 atau yang lebih baru, atau Elasticsearch 6.7 atau yang lebih baru, pada domain lokal. Versi domain jarak jauh harus lebih rendah atau versi utama yang sama dengan domain lokal. Versi Elasticsearch dianggap lebih rendah dari OpenSearch versi, artinya Anda dapat mengindeks ulang data dari domain Elasticsearch ke domain. OpenSearch Dalam versi utama yang sama, domain jarak jauh dapat berupa versi minor. Misalnya, pengindeksan ulang jarak jauh dari Elasticsearch 7.10.x ke 7.9 didukung, tetapi OpenSearch 1.0 ke Elasticsearch 7.10.x tidak didukung.

Dokumentasi lengkap untuk `reindex` operasi, termasuk langkah-langkah terperinci dan opsi yang didukung, tersedia dalam OpenSearchdokumentasi.

Topik

- Prasyarat
- Mengindeks ulang data antara domain internet OpenSearch Layanan
- Mengindeks ulang data antara domain OpenSearch Layanan saat remote berada di VPC
- Mengindeks ulang data antara domain OpenSearch non-Layanan

- Indeks ulang set data besar
- Pengaturan indeks ulang Jarak Jauh

Prasyarat

Remote reindex memiliki persyaratan sebagai berikut:

- Domain jarak jauh harus dapat diakses dari domain lokal. Untuk domain jarak jauh yang berada dalam VPC, domain lokal harus memiliki akses ke VPC. Proses ini bervariasi menurut konfigurasi jaringan, tetapi kemungkinan melibatkan koneksi ke VPN atau jaringan terkelola, atau menggunakan koneksi titik akhir VPC asli. Untuk mempelajari selengkapnya, lihat the section called “Dukungan VPC”.
- Permintaan harus disahkan oleh domain jarak jauh seperti permintaan REST lainnya. Jika domain jarak jauh telah mengaktifkan kontrol akses berbutir halus, Anda harus memiliki izin untuk melakukan reindex pada domain jarak jauh dan membaca indeks pada domain lokal. Untuk pertimbangan keamanan lebih lanjut, lihatthe section called “Kontrol akses detail”.
- Kami sarankan Anda membuat indeks dengan pengaturan yang diinginkan pada domain lokal Anda sebelum Anda memulai proses mengindeks ulang.
- Jika domain Anda menggunakan tipe instans T2 atau T3 untuk node data Anda, Anda tidak dapat menggunakan indeks ulang jarak jauh.

Mengindeks ulang data antara domain internet OpenSearch Layanan

Skenario paling dasar adalah indeks jarak jauh Wilayah AWS sama dengan domain lokal Anda dengan titik akhir yang dapat diakses publik dan Anda telah menandatangani kredensi IAM.

Dari domain jarak jauh, tentukan indeks jarak jauh untuk diindeks ulang dan indeks lokal untuk diindeks ulang ke:

```
POST _reindex
{
  "source": {
    "remote": {
      "host": "https://remote-domain-endpoint:443"
    },
    "index": "remote_index"
  },
```

```
  "dest": {
    "index": "local_index"
  }
}
```

Anda harus menambahkan 443 pada titik akhir domain jarak jauh untuk pemeriksaan validasi.

Untuk memverifikasi bahwa indeks disalin ke domain lokal, kirim permintaan ini ke domain lokal:

```
GET local_index/_search
```

Jika indeks jarak jauh berada di Wilayah yang berbeda dari domain lokal Anda, teruskan nama Regionalnya, seperti dalam permintaan sampel ini:

```
POST _reindex
{
  "source": {
    "remote": {
      "host": "https://remote-domain-endpoint:443",
      "region": "eu-west-1"
    },
    "index": "remote_index"
  },
  "dest": {
    "index": "local_index"
  }
}
```

Dalam kasus Wilayah seperti AWS GovCloud (US) atau Wilayah China yang terisolasi, titik akhir mungkin tidak dapat diakses karena pengguna IAM Anda tidak dikenali di Wilayah tersebut.

Jika domain jarak jauh diamankan dengan otentikasi dasar, tentukan nama pengguna dan kata sandi:

```
POST _reindex
{
  "source": {
    "remote": {
      "host": "https://remote-domain-endpoint:443",
      "username": "username",
      "password": "password"
    },
    "index": "remote_index"
```

```
  },
  "dest": {
    "index": "local_index"
  }
}
```

Mengindeks ulang data antara domain OpenSearch Layanan saat remote berada di VPC

Setiap domain OpenSearch Layanan terdiri dari infrastruktur internal virtual private cloud (VPC) sendiri. Saat Anda membuat domain baru di VPC OpenSearch Layanan yang ada, sebuah elastic network interface dibuat untuk setiap node data di VPC.

Karena operasi reindex jarak jauh dilakukan dari domain OpenSearch Layanan jarak jauh, dan oleh karena itu dalam VPC pribadinya sendiri, Anda memerlukan cara untuk mengakses VPC domain lokal. Anda dapat melakukan ini dengan menggunakan fitur koneksi titik akhir VPC bawaan untuk membuat koneksi melalui AWS PrivateLink, atau dengan mengonfigurasi proxy.

Jika domain lokal Anda menggunakan OpenSearch versi 1.0 atau yang lebih baru, Anda dapat menggunakan konsol atau AWS CLI untuk membuat AWS PrivateLink koneksi. AWS PrivateLink Koneksi memungkinkan sumber daya di VPC lokal untuk terhubung secara pribadi ke sumber daya di VPC jarak jauh dalam hal yang sama. Wilayah AWS

Mengindeks ulang data dengan AWS Management Console

Anda dapat menggunakan indeks ulang jarak jauh dengan konsol untuk menyalin indeks antara dua domain yang berbagi koneksi titik akhir VPC.

1. Arahkan ke konsol OpenSearch Layanan Amazon dihttps://console.aws.amazon.com/aos/.
2. Di panel navigasi kiri, pilih Domain.
3. Pilih domain lokal, atau domain yang ingin Anda salin datanya. Ini membuka halaman detail domain. Pilih tab Koneksi di bawah informasi umum dan pilih Permintaan.
4. Pada halaman Minta koneksi, pilih Koneksi Titik Akhir VPC untuk mode koneksi Anda dan masukkan detail relevan lainnya. Detail ini termasuk domain jarak jauh, yang merupakan domain yang ingin Anda salin datanya. Kemudian, pilih Permintaan.
5. Arahkan ke halaman detail domain jarak jauh, pilih tab Koneksi, dan temukan tabel Koneksi masuk. Pilih kotak centang di sebelah nama domain tempat Anda baru saja membuat koneksi dari (domain lokal). Pilih Menyetujui.

6. Arahkan kembali ke domain lokal, pilih tab Koneksi, dan temukan tabel Koneksi keluar. Setelah koneksi antara dua domain aktif, titik akhir menjadi tersedia di kolom Endpoint dalam tabel. Salin titik akhir.
7. Buka dasbor untuk domain lokal dan pilih Dev Tools di navigasi kiri. Untuk mengonfirmasi bahwa indeks domain jarak jauh belum ada di domain lokal Anda, jalankan permintaan GET berikut. Ganti *remote-domain-index-name*dengan nama indeks Anda sendiri.

```
GET remote-domain-index-name/_search
{
   "query":{
      "match_all":{}
   }
}
```

   Dalam output, Anda akan melihat kesalahan yang menunjukkan bahwa indeks tidak ditemukan.
8. Di bawah permintaan GET Anda, buat permintaan POST dan gunakan titik akhir Anda sebagai host jarak jauh, sebagai berikut.

```
POST _reindex
{
   "source":{
      "remote":{
         "host":"endpoint",
         "username":"username",
         "password":"password"
      },
      "index":"remote-domain-index-name"
   },
   "dest":{
      "index":"local-domain-index-name"
   }
}
```

   Jalankan permintaan ini.
9. Jalankan permintaan GET lagi. Output sekarang harus menunjukkan bahwa indeks lokal ada. Anda dapat melakukan kueri indeks ini untuk memverifikasi bahwa OpenSearch menyalin semua data dari indeks jarak jauh.

Mengindeks ulang data dengan operasi API OpenSearch Layanan

Anda dapat menggunakan indeks ulang jarak jauh dengan API untuk menyalin indeks antara dua domain yang berbagi koneksi titik akhir VPC.

1. Gunakan operasi CreateOutboundConnectionAPI untuk meminta koneksi baru dari domain lokal Anda ke domain jarak jauh Anda.

```
POST https://es.region.amazonaws.com/2021-01-01/opensearch/cc/outboundConnection

{
   "ConnectionAlias": "remote-reindex-example",
   "ConnectionMode": "VPC_ENDPOINT",
   "LocalDomainInfo": {
      "AWSDomainInformation": {
         "DomainName": "local-domain-name",
         "OwnerId": "aws-account-id",
         "Region": "region"
      }
   },
   "RemoteDomainInfo": {
      "AWSDomainInformation": {
         "DomainName": "remote-domain-name",
         "OwnerId": "aws-account-id",
         "Region": "region"
      }
   }
}
```

   Anda menerima a `ConnectionId` dalam tanggapan. Simpan ID ini untuk digunakan pada langkah berikutnya.

2. Gunakan operasi AcceptInboundConnectionAPI dengan ID koneksi Anda untuk menyetujui permintaan dari domain lokal.

```
PUT https://es.region.amazonaws.com/2021-01-01/opensearch/cc/
inboundConnection/ConnectionId/accept
```

3. Gunakan operasi DescribeOutboundConnectionsAPI untuk mengambil titik akhir untuk domain jarak jauh Anda.

```
{
```

```
    "Connections": [
        {
            "ConnectionAlias": "remote-reindex-example",
            "ConnectionId": "connection-id",
            "ConnectionMode": "VPC_ENDPOINT",
            "ConnectionProperties": {
                "Endpoint": "connection-endpoint"
            },
            ...
        }
    ]
}
```

Simpan *titik akhir koneksi* untuk digunakan pada Langkah 5.

4. Untuk mengonfirmasi bahwa indeks domain jarak jauh belum ada di domain lokal Anda, jalankan permintaan GET berikut. Ganti *remote-domain-index-name*dengan nama indeks Anda sendiri.

```
GET local-domain-endpoint/remote-domain-index-name/_search
{
   "query":{
      "match_all":{}
   }
}
```

Dalam output, Anda akan melihat kesalahan yang menunjukkan bahwa indeks tidak ditemukan.

5. Buat permintaan POST dan gunakan endpoint Anda sebagai host jarak jauh, sebagai berikut.

```
POST local-domain-endpoint/_reindex
{
   "source":{
      "remote":{
         "host":"connection-endpoint",
         "username":"username",
         "password":"password"
      },
      "index":"remote-domain-index-name"
   },
   "dest":{
      "index":"local-domain-index-name"
   }
```

```
}
```

Jalankan permintaan ini.

6. Jalankan permintaan GET lagi. Output sekarang harus menunjukkan bahwa indeks lokal ada. Anda dapat melakukan kueri indeks ini untuk memverifikasi bahwa OpenSearch menyalin semua data dari indeks jarak jauh.

Jika domain jarak jauh di-host di dalam VPC dan Anda tidak ingin menggunakan fitur koneksi titik akhir VPC, Anda harus mengonfigurasi proxy dengan titik akhir yang dapat diakses publik. Dalam hal ini, OpenSearch Layanan memerlukan titik akhir publik karena tidak memiliki kemampuan untuk mengirim lalu lintas ke VPC Anda.

Saat Anda menjalankan domain dalam mode VPC, satu atau beberapa titik akhir ditempatkan di VPC Anda. Namun, titik akhir ini hanya untuk lalu lintas yang masuk ke domain dalam VPC, dan mereka tidak mengizinkan lalu lintas ke VPC itu sendiri.

Perintah reindex jarak jauh dijalankan dari domain lokal, sehingga lalu lintas yang berasal tidak dapat menggunakan titik akhir tersebut untuk mengakses domain jarak jauh. Itu sebabnya proxy diperlukan dalam kasus penggunaan ini. Domain proksi harus memiliki sertifikat yang ditandatangani oleh otoritas sertifikat publik (CA). Sertifikat yang ditandatangani sendiri atau ditandatangani CA privat tidak didukung.

Mengindeks ulang data antara domain OpenSearch non-Layanan

Jika indeks jarak jauh di-host di luar OpenSearch Layanan, seperti pada instans EC2 yang dikelola sendiri, setel `external` parameter ke: `true`

```
POST _reindex
{
  "source": {
    "remote": {
      "host": "https://remote-domain-endpoint:443",
      "username": "username",
      "password": "password",
      "external": true
    },
    "index": "remote_index"
  },
  "dest": {
    "index": "local_index"
```

```
  }
}
```

Dalam hal ini, hanya otentikasi dasar dengan nama pengguna dan kata sandi yang didukung. Domain jarak jauh harus memiliki titik akhir yang dapat diakses publik (meskipun berada di VPC yang sama dengan domain OpenSearch Layanan lokal) dan sertifikat yang ditandatangani oleh CA publik. Sertifikat yang ditandatangani sendiri atau ditandatangani CA pribadi tidak didukung.

Indeks ulang set data besar

Indeks ulang jarak jauh mengirimkan permintaan gulir ke domain jarak jauh dengan nilai default berikut:

- Konteks pencarian 5 menit
- Batas waktu soket 30 detik
- Ukuran Batch 1.000

Kami merekomendasikan untuk menyetel parameter ini untuk mengakomodasi data Anda. Untuk dokumen besar, pertimbangkan ukuran batch yang lebih kecil dan/atau batas waktu yang lebih lama. Untuk informasi selengkapnya, lihat Pencarian Gulir.

```
POST _reindex?pretty=true&scroll=10h&wait_for_completion=false
{
  "source": {
    "remote": {
      "host": "https://remote-domain-endpoint:443",
      "socket_timeout": "60m"
    },
    "size": 100,
    "index": "remote_index"
  },
  "dest": {
    "index": "local_index"
  }
}
```

Kami juga merekomendasikan menambahkan pengaturan berikut ke indeks lokal untuk performa yang lebih baik:

```
PUT local_index
```

```
{
  "settings": {
    "refresh_interval": -1,
    "number_of_replicas": 0
  }
}
```

Setelah proses indeks ulang selesai, Anda dapat mengatur jumlah replika yang diinginkan dan menghapus pengaturan interval refresh.

Untuk mengindeks ulang hanya sebagian dokumen yang Anda pilih melalui kueri, kirim permintaan ini ke domain lokal:

```
POST _reindex
{
  "source": {
    "remote": {
      "host": "https://remote-domain-endpoint:443"
    },
    "index": "remote_index",
    "query": {
      "match": {
        "field_name": "text"
      }
    }
  },
  "dest": {
    "index": "local_index"
  }
}
```

Pengindeksan ulang jarak jauh tidak mendukung pemotongan, jadi Anda tidak dapat melakukan beberapa operasi gulir untuk permintaan yang sama secara paralel.

Pengaturan indeks ulang Jarak Jauh

Selain opsi pengindeksan ulang standar, OpenSearch Layanan mendukung opsi berikut:

Opsi	Nilai yang valid	Deskripsi	Wajib
eksternal	Boolean	Jika domain jarak jauh bukan domain	Tidak

Opsi	Nilai yang valid	Deskripsi	Wajib
		OpenSearch Layanan, atau jika Anda mengindeks ulang antara dua domain VPC, tentukan sebagai. `true`	
wilayah	String	Jika domain jarak jauh berada di Wilayah yang berbeda, tentukan nama Wilayah.	Tidak

Mengelola data deret waktu di Amazon OpenSearch Service dengan aliran data

Alur kerja tipikal untuk mengelola data deret waktu melibatkan beberapa langkah, seperti membuat alias indeks rollover, menentukan indeks tulis, dan menentukan pemetaan dan pengaturan umum untuk indeks pendukung.

Aliran data di Amazon OpenSearch Service membantu menyederhanakan proses penyiapan awal ini. Aliran data bekerja di luar kotak untuk data berbasis waktu seperti log aplikasi yang biasanya hanya ditambahkan.

Aliran data membutuhkan OpenSearch 1.0 atau lebih baru. Dokumentasi lengkap untuk fitur tersedia dalam OpenSearchdokumentasi.

Memulai dengan aliran data

Aliran data secara internal terdiri dari beberapa indeks dukungan. Permintaan pencarian dialihkan ke semua indeks dukungan, sementara permintaan pengindeksan dialihkan ke indeks penulisan terbaru.

Langkah 1: Buat template indeks

Untuk membuat aliran data, Anda harus terlebih dahulu membuat template indeks yang mengkonfigurasi sekumpulan indeks sebagai aliran data. `data_stream`Objek menunjukkan bahwa itu adalah aliran data dan bukan template indeks biasa. Pola indeks cocok dengan nama aliran data:

```
PUT _index_template/logs-template
{
  "index_patterns": [
    "my-data-stream",
    "logs-*"
  ],
  "data_stream": {},
  "priority": 100
}
```

Dalam hal ini, setiap dokumen yang dicerna harus memiliki `@timestamp` bidang. Anda juga dapat menentukan bidang timestamp kustom Anda sendiri sebagai properti di `data_stream` objek:

```
PUT _index_template/logs-template
{
  "index_patterns": "my-data-stream",
  "data_stream": {
    "timestamp_field": {
      "name": "request_time"
    }
  }
}
```

Langkah 2: Buat aliran data

Setelah Anda membuat template indeks, Anda dapat langsung mulai menelan data tanpa membuat aliran data.

Karena kita memiliki template indeks yang cocok dengan `data_stream` objek, OpenSearch secara otomatis membuat aliran data:

```
POST logs-staging/_doc
{
  "message": "login attempt failed",
  "@timestamp": "2013-03-01T00:00:00"
```

```
}
```

Langkah 3: Menelan data ke dalam aliran data

Untuk menyerap data ke dalam aliran data, Anda dapat menggunakan API pengindeksan reguler. Pastikan setiap dokumen yang Anda indeks memiliki bidang timestamp. Jika Anda mencoba untuk mengambil dokumen yang tidak memiliki kolom stempel waktu, Anda mendapatkan pesan kesalahan.

```
POST logs-redis/_doc
{
  "message": "login attempt",
  "@timestamp": "2013-03-01T00:00:00"
}
```

Langkah 4: Mencari aliran data

Anda dapat mencari aliran data seperti Anda mencari indeks biasa atau alias indeks. Operasi pencarian berlaku untuk semua indeks backing (semua data yang ada di sungai).

```
GET logs-redis/_search
{
  "query": {
    "match": {
      "message": "login"
    }
  }
}
```

Langkah 5: Rollover aliran data

Anda dapat mengatur kebijakan Index State Management (ISM) untuk mengotomatiskan proses rollover untuk aliran data. Kebijakan ISM diterapkan pada indeks pendukung pada saat penciptaannya. Saat Anda mengaitkan kebijakan ke aliran data, itu hanya memengaruhi indeks dukungan aliran data tersebut di future. Anda juga tidak perlu menyediakan `rollover_alias` pengaturan, karena kebijakan ISM menyimpulkan informasi ini dari indeks dukungan.

> **Note**
>
> Jika Anda memigrasi indeks dukungan ke penyimpanan dingin, OpenSearch hapus indeks ini dari aliran data. Bahkan jika Anda memindahkan indeks kembali ke UltraWarm, indeks tetap

independen dan bukan bagian dari aliran data asli. Setelah indeks dihapus dari aliran data, pencarian terhadap aliran tidak akan mengembalikan data apa pun dari indeks.

Warning

Indeks tulis untuk aliran data tidak dapat dimigrasi ke cold storage. Jika Anda ingin memigrasi data dalam aliran data ke cold storage, Anda harus menggulingkan aliran data sebelum migrasi.

Langkah 6: Kelola aliran data di Dasbor OpenSearch

Untuk mengelola aliran data dari OpenSearch Dasbor, buka OpenSearchDasbor, pilih Manajemen Indeks, pilih Indeks atau Indeks yang dikelola kebijakan.

Langkah 7: Menghapus aliran data

Operasi delete pertama menghapus indeks backing dari aliran data dan kemudian menghapus aliran data itu sendiri.

Untuk menghapus aliran data dan semua indeks pendukungnya yang tersembunyi:

```
DELETE _data_stream/name_of_data_stream
```

Pemantauan data di Amazon OpenSearch Layanan

Pemantauan data Anda secara proaktif di Amazon OpenSearch Layanan dengan deteksi peringatan dan anomali. Atur peringatan untuk menerima pemberitahuan bila data Anda melebihi ambang batas tertentu. Deteksi anomali menggunakan machine learning untuk secara otomatis mendeteksi adanya outlier dalam data streaming Anda. Anda dapat memasangkan deteksi anomali dengan pemberitahuan untuk memastikan Anda diberitahu dengan segera setelah anomali terdeteksi.

Topik

- Mengonfigurasi peringatan di Layanan Amazon OpenSearch
- Deteksi anomali di Amazon OpenSearch Service

Mengonfigurasi peringatan di Layanan Amazon OpenSearch

Konfigurasikan peringatan di OpenSearch Layanan Amazon untuk mendapatkan pemberitahuan saat data dari satu atau beberapa indeks memenuhi persyaratan tertentu. Misalnya, Anda mungkin ingin menerima email jika aplikasi Anda mencatat lebih dari lima kesalahan HTTP 503 dalam satu jam, atau Anda mungkin ingin halaman developer jika tidak ada dokumen baru yang diindeks dalam 20 menit terakhir.

Peringatan membutuhkan OpenSearch atau Elasticsearch 6.2 atau yang lebih baru. Untuk dokumentasi lengkap, termasuk deskripsi API, lihat Peringatan dalam dokumentasi. OpenSearch Topik ini menyoroti perbedaan peringatan di OpenSearch Layanan dibandingkan dengan versi open source.

Topik

- Izin peringatan
- Memulai dengan peringatan
- Notifikasi
- Perbedaan

Izin peringatan

Peringatan mendukung kontrol akses yang sangat baik. Untuk detail tentang pencampuran dan pencocokan izin agar sesuai dengan kasus penggunaan Anda, lihat Memperingatkan keamanan dalam dokumentasi. OpenSearch

Untuk mengakses halaman Peringatan dalam OpenSearch Dasbor, Anda setidaknya harus dipetakan ke peran yang `alerting_read_access` telah ditentukan sebelumnya, atau diberikan izin yang setara. Peran ini memberikan izin untuk melihat peringatan, tujuan, dan monitor, tetapi tidak untuk mengakui peringatan atau memodifikasi tujuan atau monitor.

Memulai dengan peringatan

Untuk membuat peringatan, Anda mengonfigurasi monitor, yang merupakan pekerjaan yang berjalan pada jadwal yang ditentukan dan OpenSearch indeks kueri. Anda juga mengonfigurasi satu atau beberapa pemicu, yang menentukan kondisi yang menghasilkan peristiwa. Terakhir, Anda mengonfigurasi tindakan, yang terjadi setelah peringatan dipicu.

Untuk memulai peringatan

1. Pilih Peringatan dari menu utama OpenSearch Dasbor dan pilih Buat monitor.
2. Buat monitor per kueri, per-bucket, per-cluster, atau per-dokumen. Untuk petunjuk, lihat Membuat monitor.
3. Untuk Pemicu, buat satu atau lebih pemicu. Untuk petunjuk, lihat Membuat pemicu.
4. Untuk Tindakan, siapkan saluran notifikasi untuk peringatan tersebut. Pilih antara Slack, Amazon Chime, webhook khusus, atau Amazon SNS. Seperti yang Anda bayangkan, notifikasi memerlukan konektivitas ke saluran. Misalnya, domain OpenSearch Layanan Anda harus dapat terhubung ke internet untuk memberi tahu saluran Slack atau mengirim webhook khusus ke server pihak ketiga. Webhook kustom harus memiliki alamat IP publik agar domain OpenSearch Layanan dapat mengirim peringatan kepadanya.

> **Tip**
>
> Setelah tindakan berhasil mengirim pesan, mengamankan akses ke pesan tersebut (misalnya, akses ke saluran Slack) adalah tanggung jawab Anda. Jika domain Anda berisi data sensitif, pertimbangkan untuk menggunakan pemicu tanpa tindakan dan secara berkala memeriksa Dashboard untuk peringatan.

Notifikasi

Peringatan terintegrasi dengan Notifikasi, yang merupakan sistem terpadu untuk notifikasi. OpenSearch Pemberitahuan memungkinkan Anda mengonfigurasi layanan komunikasi mana yang ingin Anda gunakan dan melihat statistik dan informasi pemecahan masalah yang relevan. Untuk dokumentasi yang komprehensif, lihat Pemberitahuan dalam OpenSearch dokumentasi.

Domain Anda harus menjalankan OpenSearch versi 2.3 atau yang lebih baru untuk menggunakan notifikasi.

> **Note**
>
> OpenSearch pemberitahuan terpisah dari pemberitahuan OpenSearch Layanan, yang memberikan rincian tentang pembaruan perangkat lunak layanan, penyempurnaan Auto-Tune, dan informasi tingkat domain penting lainnya. OpenSearch notifikasi khusus plugin.

Saluran notifikasi menggantikan tujuan peringatan dimulai dengan OpenSearch versi 2.0. Tujuan secara resmi tidak digunakan lagi, dan semua pemberitahuan peringatan akan dikelola melalui saluran ke depan.

Saat Anda memutakhirkan domain ke versi 2.3 atau yang lebih baru (karena dukungan OpenSearch Layanan untuk 2.x dimulai dengan 2.3), tujuan yang ada akan dimigrasikan secara otomatis ke saluran notifikasi. Jika tujuan gagal bermigrasi, monitor akan terus menggunakannya hingga monitor dimigrasikan ke saluran notifikasi. Untuk informasi lebih lanjut, lihat Pertanyaan tentang tujuan dalam dokumentasi. OpenSearch

Untuk memulai notifikasi, masuk ke OpenSearch Dasbor dan pilih Notifikasi, Saluran, dan Buat saluran.

Amazon Simple Notification Service (Amazon SNS) adalah jenis saluran yang didukung untuk notifikasi. Untuk mengautentikasi pengguna, Anda harus memberi pengguna akses penuh ke Amazon SNS, atau membiarkan mereka mengambil peran IAM yang memiliki izin untuk mengakses Amazon SNS. Untuk petunjuk, lihat Amazon SNS sebagai jenis saluran.

Perbedaan

Dibandingkan dengan versi open-source OpenSearch, peringatan di Amazon OpenSearch Service memiliki beberapa perbedaan penting.

Pengaturan peringatan

OpenSearch Layanan memungkinkan Anda mengubah pengaturan peringatan berikut:

- `plugins.scheduled_jobs.enabled`
- `plugins.alerting.alert_history_enabled`
- `plugins.alerting.alert_history_max_age`
- `plugins.alerting.alert_history_max_docs`
- `plugins.alerting.alert_history_retention_period`
- `plugins.alerting.alert_history_rollover_period`
- `plugins.alerting.filter_by_backend_roles`

Semua pengaturan lainnya menggunakan nilai default yang tidak dapat Anda ubah.

Untuk menonaktifkan peringatan, kirim permintaan berikut:

```
PUT _cluster/settings
{
  "persistent" : {
    "plugins.scheduled_jobs.enabled" : false
  }
}
```

Permintaan berikut mengonfigurasi peringatan untuk secara otomatis menghapus indeks riwayat setelah tujuh hari, bukan default 30 hari:

```
PUT _cluster/settings
{
  "persistent": {
    "plugins.alerting.alert_history_retention_period": "7d"
  }
}
```

Jika sebelumnya Anda membuat monitor dan ingin menghentikan pembuatan indeks peringatan harian, hapus semua indeks riwayat peringatan:

```
DELETE .plugins-alerting-alert-history-*
```

Untuk mengurangi hitungan serpihan untuk indeks riwayat, buat template indeks. Permintaan berikut menetapkan indeks riwayat untuk peringatan ke satu pecahan dan satu replika:

```
PUT _index_template/template-name
{
  "index_patterns": [".opendistro-alerting-alert-history-*"],
  "template": {
    "settings": {
      "number_of_shards": 1,
      "number_of_replicas": 1
    }
  }
}
```

Tergantung pada toleransi kehilangan data, Anda mungkin dapat mempertimbangkan untuk menggunakan nol replika. Untuk informasi selengkapnya tentang membuat dan mengelola templat indeks, lihat Templat indeks dalam OpenSearch dokumentasi.

Deteksi anomali di Amazon OpenSearch Service

Deteksi anomali di Amazon OpenSearch Service secara otomatis mendeteksi anomali dalam OpenSearch data Anda dalam waktu dekat dengan menggunakan algoritme Random Cut Forest (RCF). RCF adalah algoritme machine learning tanpa pengawasan yang memodelkan sketsa aliran data masuk Anda. Algoritma menghitung`anomaly grade` dan`confidence score` nilai untuk setiap titik data yang masuk. Deteksi anomali menggunakan nilai ini untuk membedakan anomali dari variasi normal dalam data Anda.

Anda dapat memasangkan plugin deteksi anomali dengan plugin the section called “Peringatan” untuk memberitahu Anda segera setelah anomali terdeteksi.

Deteksi anomali tersedia di domain yang menjalankan OpenSearch versi Elasticsearch 7.4 atau yang lebih baru. Semua tipe instans mendukung deteksi anomali kecuali untuk `t2.micro` dan `t2.small`. Dokumentasi lengkap untuk deteksi anomali, termasuk langkah-langkah terperinci dan deskripsi API, tersedia di OpenSearchdokumentasi.

Prasyarat

Deteksi anomali memiliki prasyarat berikut:

- Deteksi anomali memerlukan OpenSearch Elasticsearch 7.4 atau yang lebih baru.

- Deteksi anomali hanya mendukung kontrol akses berbutir halus pada Elasticsearch versi 7.9 dan yang lebih baru dan semua versi OpenSearch. Sebelum Elasticsearch 7.9, hanya pengguna admin yang dapat membuat, melihat, dan mengelola detektor.
- Jika domain Anda menggunakan kontrol akses berbutir halus, pengguna non-admin harus dipetakan `keanomaly_read_access` peran di OpenSearch Dasbor untuk melihat detektor, atau`anomaly_full_access` untuk membuat dan mengelola detektor.

Memulai dengan deteksi anomali

Untuk memulai, pilih Deteksi Anomali di OpenSearch Dasbor.

Langkah 1: Buat detektor

Detektor adalah tugas deteksi anomali individu. Anda dapat membuat beberapa detektor, dan semua detektor dapat berjalan secara bersamaan, dengan masing-masing menganalisis data dari sumber yang berbeda.

Langkah 2: Tambahkan fitur ke detektor

Sebuah fitur adalah bidang dalam indeks Anda yang Anda memeriksa anomali. Detektor dapat menemukan anomali di satu atau beberapa fitur. Anda harus memilih salah satu agregasi berikut untuk setiap fitur:`average()`, `sum()`, `count()`, `min()`, atau `max()`.

> **Note**
>
> Metode`count()` agregasi hanya tersedia di OpenSearch Elasticsearch 7.7 atau yang lebih baru. Untuk Elasticsearch 7.4, gunakan ekspresi kustom seperti berikut ini:
>
> ```
> {
>   "aggregation_name": {
>      "value_count": {
>         "field": "field_name"
>      }
>   }
> }
> ```

Metode agregasi menentukan apa yang merupakan anomali. Misalnya, jika Anda memilih `min()`, detektor berfokus pada menemukan anomali berdasarkan nilai minimum fitur Anda. Jika Anda

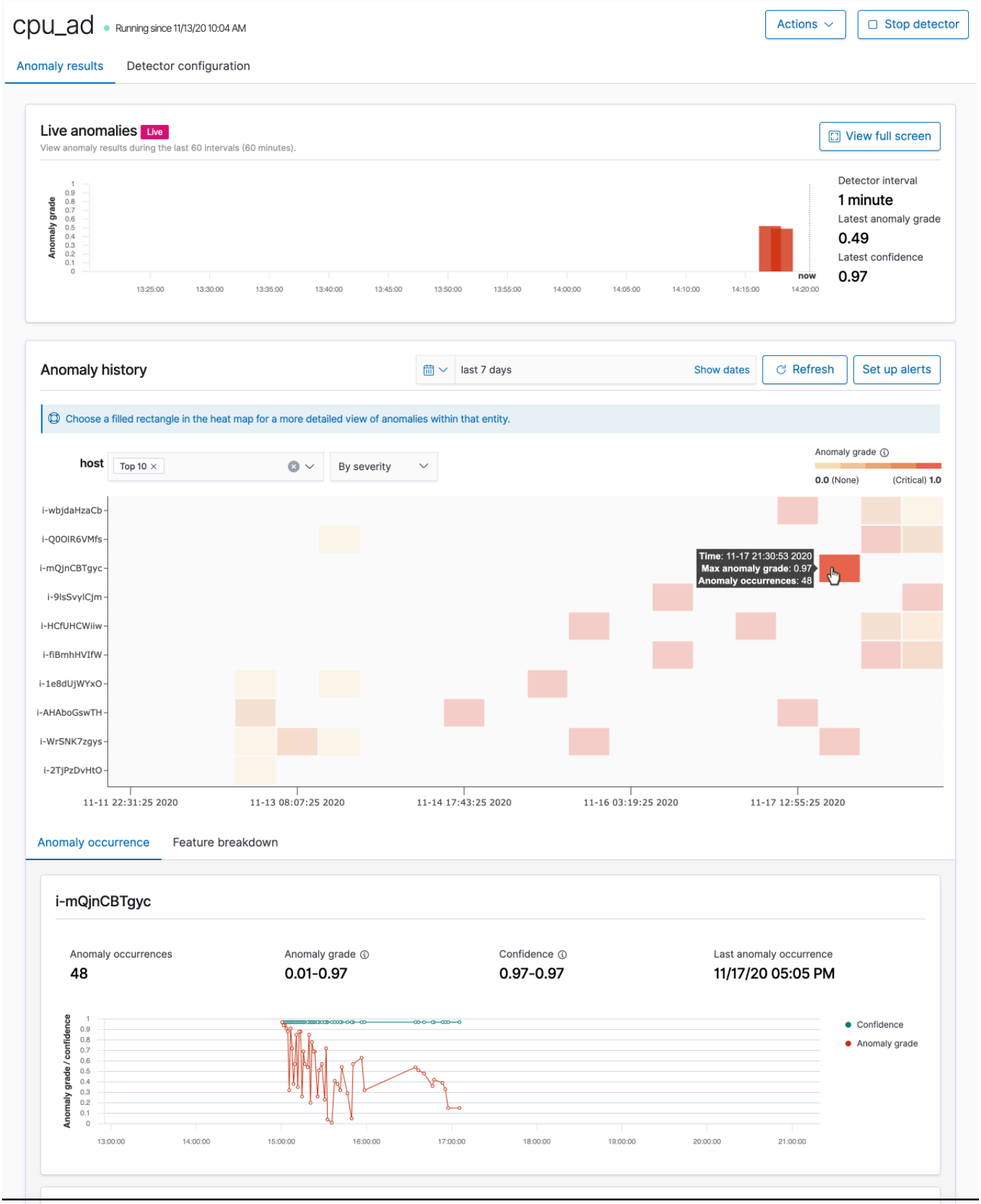
memilih `average()`, detektor menemukan anomali berdasarkan nilai rata-rata fitur Anda. Anda dapat menambahkan maksimum lima fitur per detektor.

Anda dapat mengonfigurasi pengaturan opsional berikut (tersedia di Elasticsearch 7.7 dan yang lebih baru):

- Bidang kategori - Mengategorikan atau mengiris data Anda dengan dimensi seperti alamat IP, ID produk, kode negara, dan sebagainya.
- Ukuran jendela - Mengatur jumlah interval agregasi dari aliran data Anda untuk mempertimbangkan dalam jendela deteksi.

Setelah mengatur fitur, pratinjau anomali sampel dan sesuaikan pengaturan fitur jika perlu.

Langkah 3: Perhatikan hasilnya



- Anomali hidup - menampilkan hasil anomali hidup untuk 60 interval terakhir. Misalnya, jika interval diatur ke 10, itu menunjukkan hasil untuk 600 menit terakhir. Grafik ini di-refresh setiap 30 detik.
- Riwayat anomali - plot kelas anomali dengan ukuran kepercayaan yang sesuai.
- Perincian fitur - plot fitur berdasarkan metode agregasi. Anda dapat memvariasikan rentang tanggal-waktu detektor.
- Kejadian anomali - menunjukkan `Start time`, `End time`, `Data confidence`, dan `Anomaly grade` untuk setiap anomali terdeteksi.

  Jika Anda menetapkan bidang kategori, Anda akan melihat bagan Peta panas tambahan yang berkorelasi hasil untuk entitas anomali. Pilih persegi panjang yang terisi untuk melihat tampilan anomali yang lebih rinci.

Untuk Langkah 4: Menyiapkan peringatan

Untuk membuat monitor untuk mengirimkan pemberitahuan bila ada anomali yang terdeteksi, pilih Menyiapkan peringatan. Plugin mengarahkan Anda ke halaman Tambahkan monitor di mana dapat mengonfigurasi peringatan.

Tutorial: Mendeteksi penggunaan CPU yang tinggi dengan deteksi anomali

Tutorial ini menunjukkan cara membuat detektor anomali di Amazon OpenSearch Service untuk mendeteksi penggunaan CPU yang tinggi. Anda akan menggunakan OpenSearch Dasbor untuk mengonfigurasi detektor untuk memantau penggunaan CPU, dan menghasilkan peringatan saat penggunaan CPU Anda naik di atas ambang batas yang ditentukan.

> **Note**
>
> Langkah-langkah ini berlaku untuk versi terbaru OpenSearch dan mungkin sedikit berbeda untuk versi sebelumnya.

Prasyarat

- Anda harus memiliki domain OpenSearch Layanan yang menjalankan Elasticsearch 7.4 atau versi yang lebih baru, atau OpenSearch versi apa pun.
- Anda harus menelan file log aplikasi ke dalam klaster Anda yang berisi data penggunaan CPU.

Langkah 1: Buat detektor

Pertama, buat detektor yang mengidentifikasi anomali dalam data penggunaan CPU Anda.

1. Buka menu panel kiri di OpenSearch Dasbor dan pilih Deteksi Anomali, lalu pilih Buat detektor.
2. Beri nama detektor**`high-cpu-usage`**.
3. Untuk sumber data Anda, pilih indeks yang berisi file log penggunaan CPU tempat Anda ingin mengidentifikasi anomali.
4. Pilih bidang Timestamp dari data Anda. Opsional, Anda dapat menambahkan filter data. Filter data ini hanya menganalisis subset dari sumber data dan mengurangi kebisingan dari data yang tidak relevan.
5. Atur interval Detector menjadi 2 menit. Interval ini menentukan waktu (dengan interval menit) bagi detektor untuk mengumpulkan data.
6. Di Window delay, tambahkan penundaan 1 menit. Penundaan ini menambah waktu pemrosesan ekstra untuk memastikan bahwa semua data dalam jendela hadir.
7. Pilih Selanjutnya. Pada dasbor deteksi anomali, di bawah nama detektor, pilih Konfigurasi model.
8. Untuk nama Fitur, masukkan**`max_cpu_usage`**. Untuk Status fitur, pilih Aktifkan fitur.
9. Untuk Temukan anomali berdasarkan, pilih Nilai bidang.
10. Untuk metode Agregasi, pilih **`max()`**.
11. Untuk bidang, pilih bidang di data Anda untuk memeriksa anomali. Misalnya, mungkin disebut`cpu_usage_percentage`.
12. Simpan semua pengaturan lainnya sebagai default dan pilih Berikutnya.
13. Abaikan pengaturan pekerjaan detektor dan pilih Berikutnya.
14. Di jendela pop-up, pilih kapan harus memulai detektor (secara otomatis atau manual), lalu pilih Konfigurasi.

Sekarang detektor dikonfigurasi, setelah diinisialisasi, Anda akan dapat melihat hasil real-time dari penggunaan CPU di bagian hasil Real-time pada panel detektor Anda. Bagian anomali Live menampilkan anomali apa pun yang terjadi saat data sedang dicerna secara real time.

Langkah 2: Konfigurasi peringatan

Sekarang setelah Anda membuat detektor, buat monitor yang memanggil peringatan untuk mengirim pesan ke Slack saat mendeteksi penggunaan CPU yang memenuhi kondisi yang ditentukan dalam

pengaturan detektor. Anda akan menerima notifikasi Slack saat data dari satu atau beberapa indeks memenuhi ketentuan yang memanggil peringatan.

1. Buka menu panel kiri di OpenSearch Dasbor dan pilih Peringatan, lalu pilih Create monitor.
2. Berikan nama untuk monitor.
3. Untuk jenis Monitor, pilih Monitor per-query. Sebuah monitor per-query menjalankan query tertentu dan mendefinisikan pemicu.
4. Untuk metode penentuan Monitor, pilih detektor Anomali, lalu pilih detektor yang Anda buat di bagian sebelumnya dari menu dropdown Detector.
5. Untuk Jadwal, pilih seberapa sering monitor mengumpulkan data dan seberapa sering Anda menerima peringatan. Untuk keperluan tutorial ini, atur jadwal untuk dijalankan setiap 7 menit.
6. Di bagian Pemicu, pilih Tambahkan pemicu. Untuk nama Trigger, masukkan**`High CPU usage`**. Untuk tutorial ini, untuk tingkat keparahan, pilih 1, yang merupakan tingkat keparahan tertinggi.
7. Untuk ambang batas kelas Anomali, pilih IS ABOVE. Pada menu di bawah itu, pilih ambang batas kelas yang akan diterapkan. Untuk tutorial ini, atur kelas Anomali menjadi 0,7.
8. Untuk ambang kepercayaan anomali, pilih IS DI ATAS. Pada menu di bawah itu, masukkan nomor yang sama dengan kelas Anomali Anda. Untuk tutorial ini, atur ambang kepercayaan Anomali menjadi 0,7.
9. Di bagian Tindakan, pilih Tujuan. Di bidang Nama, pilih nama tujuan. Pada menu Type, pilih Slack. Di bidang URL Webhook, masukkan URL webhook untuk menerima peringatan. Untuk informasi selengkapnya, lihat Pengiriman pesan menggunakan webhook masuk.
10. Pilih Create (Buat).

Sumber daya terkait

- the section called “Peringatan”

- the section called “Deteksi anomali”
- API deteksi anomali

Pembelajaran mesin untuk OpenSearch Layanan Amazon

ML Commons adalah OpenSearch plugin yang menyediakan satu set algoritma machine learning (ML) umum melalui panggilan transport dan REST API. Panggilan tersebut memilih node dan sumber daya yang tepat untuk setiap permintaan ML dan memantau tugas ML untuk memastikan waktu aktif. Hal ini memungkinkan Anda untuk memanfaatkan algoritme Open Source yang ada dan mengurangi upaya yang diperlukan untuk mengembangkan fitur ML baru. Untuk selengkapnya tentang plugin, lihat Pembelajaran mesin dalam OpenSearch dokumentasi. Bab ini membahas cara menggunakan plugin dengan Amazon OpenSearch Service.

Topik

Konektor Amazon OpenSearch Service MS untuk Layanan AWS

Saat Anda menggunakan konektor pembelajaran mesin Amazon OpenSearch Service (ML) dengan konektor lain Layanan AWS, Anda perlu menyiapkan peran IAM untuk menghubungkan OpenSearch Layanan dengan aman ke layanan tersebut. Layanan AWS bahwa Anda dapat mengatur konektor untuk menyertakan Amazon SageMaker dan Amazon Bedrock. Dalam tutorial ini, kita membahas cara membuat konektor dari OpenSearch Service ke SageMaker Runtime. Untuk informasi selengkapnya tentang konektor, lihat Konektor yang didukung.

Topik

Prasyarat

Untuk membuat konektor, Anda harus memiliki endpoint SageMaker Domain Amazon dan peran IAM yang memberikan akses Layanan OpenSearch .

Menyiapkan SageMaker Domain Amazon

Lihat Menerapkan Model di Amazon SageMaker di Panduan SageMaker Pengembang Amazon untuk menerapkan model pembelajaran mesin Anda. Perhatikan URL titik akhir untuk model Anda, yang Anda perlukan untuk membuat konektor AI.

Membuat peran IAM

Siapkan peran IAM untuk mendelegasikan izin SageMaker Runtime ke Layanan. OpenSearch Untuk membuat peran baru, lihat Membuat peran IAM (konsol) di Panduan Pengguna IAM. Secara opsional, Anda dapat menggunakan peran yang ada selama memiliki set hak istimewa yang sama. Jika Anda membuat peran baru alih-alih menggunakan peran AWS terkelola, ganti `opensearch-sagemaker-role` dalam tutorial ini dengan nama peran Anda sendiri.

1. Lampirkan kebijakan IAM terkelola berikut ke peran baru Anda untuk memungkinkan OpenSearch Layanan mengakses titik SageMaker akhir Anda. Untuk melampirkan kebijakan ke peran, lihat Menambahkan izin identitas IAM.

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

2. Ikuti instruksi dalam Memodifikasi kebijakan kepercayaan peran untuk mengedit hubungan kepercayaan peran. Anda harus menentukan OpenSearch Layanan dalam `Principal` pernyataan:

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

```
                "sts:AssumeRole"
            ],
            "Effect": "Allow",
            "Principal": {
                "Service": [
                    "opensearchservice.amazonaws.com"
                ]
            }
        }
    ]
}
```

Kami menyarankan Anda menggunakan tombol `aws:SourceAccount` dan `aws:SourceArn` kondisi untuk membatasi akses ke domain tertentu. Itu `SourceAccount` adalah Akun AWS ID milik pemilik domain, dan `SourceArn` adalah ARN dari domain. Misalnya, Anda dapat menambahkan blok kondisi berikut ke kebijakan kepercayaan:

```
"Condition": {
    "StringEquals": {
        "aws:SourceAccount": "account-id"
    },
    "ArnLike": {
        "aws:SourceArn": "arn:aws:es:region:account-id:domain/domain-name"
    }
}
```

Konfigurasi izin

Untuk membuat konektor, Anda memerlukan izin untuk meneruskan peran IAM ke OpenSearch Layanan. Anda juga memerlukan akses ke tindakan `es:ESHttpPost`. Untuk memberikan kedua izin ini, lampirkan kebijakan berikut ke peran IAM yang kredensialnya digunakan untuk menandatangani permintaan:

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Effect": "Allow",
      "Action": "iam:PassRole",
      "Resource": "arn:aws:iam::account-id:role/opensearch-sagemaker-role"
```

```
    },
    {
      "Effect": "Allow",
      "Action": "es:ESHttpPost",
      "Resource": "arn:aws:es:region:account-id:domain/domain-name/*"
    }
  ]
}
```

Jika pengguna atau peran Anda tidak memiliki `iam:PassRole` izin untuk meneruskan peran Anda, Anda mungkin mengalami kesalahan otorisasi saat mencoba mendaftarkan repositori di langkah berikutnya.

Petakan peran ML di OpenSearch Dasbor (jika menggunakan kontrol akses berbutir halus)

Kontrol akses berbutir halus memperkenalkan langkah tambahan saat menyiapkan konektor. Bahkan jika Anda menggunakan autentikasi basic HTTP untuk semua tujuan lain, Anda perlu memetakan peran `ml_full_access` ke IAM role Anda yang memiliki izin `iam:PassRole` untuk meneruskan `opensearch-sagemaker-role`.

1. Arahkan ke plugin OpenSearch Dasbor untuk domain OpenSearch Layanan Anda. Anda dapat menemukan titik akhir Dasbor di dasbor domain Anda di konsol OpenSearch Layanan.
2. Dari menu utama pilih Keamanan, Peran, dan pilih peran ml_full_access.
3. Pilih Pengguna yang Dipetakan, Kelola pemetaan.
4. Di bawah peran Backend, tambahkan ARN dari peran yang memiliki izin untuk diteruskan. `opensearch-sagemaker-role`

   ```
   arn:aws:iam::account-id:role/role-name
   ```

5. Pilih Peta dan konfirmasikan pengguna atau peran muncul di bawah Pengguna yang dipetakan.

Buat konektor OpenSearch Service

Untuk membuat konektor, kirim `POST` permintaan ke titik akhir domain OpenSearch Layanan. Anda dapat menggunakan curl, contoh klien Python, Postman, atau metode lain untuk mengirim permintaan yang ditandatangani. Perhatikan bahwa Anda tidak dapat menggunakan `POST` permintaan di konsol Kibana. Permintaan mengambil format berikut:

```
POST domain-endpoint/_plugins/_ml/connectors/_create
{
   "name": "sagemaker: embedding",
   "description": "Test connector for Sagemaker embedding model",
   "version": 1,
   "protocol": "aws_sigv4",
   "credential": {
      "roleArn": "arn:aws:iam::account-id:role/opensearch-sagemaker-role"
   },
   "parameters": {
      "region": "region",
      "service_name": "sagemaker"
   },
   "actions": [
      {
         "action_type": "predict",
         "method": "POST",
         "headers": {
            "content-type": "application/json"
         },
         "url": "https://runtime.sagemaker.region.amazonaws.com/endpoints/endpoint-id/
invocations",
         "request_body": "{ \"inputs\": { \"question\": \"${parameters.question}\",
 \"context\": \"${parameters.context}\" } }"
      }
   ]
}
```

Jika domain Anda berada dalam virtual private cloud (VPC), komputer Anda harus terhubung ke VPC agar permintaan berhasil membuat konektor AI. Mengakses VPC bervariasi menurut konfigurasi jaringan, tetapi biasanya melibatkan koneksi ke VPN atau jaringan perusahaan. Untuk memastikan bahwa Anda dapat mencapai domain OpenSearch Layanan, `https://your-vpc-domain.region.es.amazonaws.com` navigasikan ke browser web dan verifikasi bahwa Anda menerima respons JSON default.

Contoh klien Python

Klien Python lebih sederhana untuk diotomatisasi daripada permintaan HTTP dan memiliki kegunaan ulang yang lebih baik. Untuk membuat konektor AI dengan klien Python, simpan kode contoh berikut ke file Python. Klien membutuhkan AWS SDK for Python (Boto3), `requests`, dan `requests-aws4auth`paket.

```
import boto3
import requests
from requests_aws4auth import AWS4Auth

host = 'domain-endpoint/'
region = 'region'
service = 'es'
credentials = boto3.Session().get_credentials()
awsauth = AWS4Auth(credentials.access_key, credentials.secret_key, region, service,
 session_token=credentials.token)

# Register repository
path = '_plugins/_ml/connectors/_create'
url = host + path

payload = {
   "name": "sagemaker: embedding",
   "description": "Test connector for Sagemaker embedding model",
   "version": 1,
   "protocol": "aws_sigv4",
   "credential": {
      "roleArn": "arn:aws:iam::account-id:role/opensearch-sagemaker-role"
   },
   "parameters": {
      "region": "region",
      "service_name": "sagemaker"
   },
   "actions": [
      {
         "action_type": "predict",
         "method": "POST",
         "headers": {
            "content-type": "application/json"
         },
         "url": "https://runtime.sagemaker.region.amazonaws.com/endpoints/endpoint-id/
invocations",
         "request_body": "{ \"inputs\": { \"question\": \"${parameters.question}\",
 \"context\": \"${parameters.context}\" } }"
      }
   ]
}
headers = {"Content-Type": "application/json"}
```

```
r = requests.post(url, auth=awsauth, json=payload, headers=headers)
print(r.status_code)
print(r.text)
```

Konektor Amazon OpenSearch Service MS untuk platform pihak ketiga

Dalam tutorial ini, kita membahas cara membuat konektor dari OpenSearch Service to Cohere. Untuk informasi selengkapnya tentang konektor, lihat Konektor yang didukung.

Saat Anda menggunakan konektor pembelajaran mesin Amazon OpenSearch Service (ML) dengan model jarak jauh eksternal, Anda perlu menyimpan kredensial otorisasi spesifik Anda. AWS Secrets Manager Ini bisa berupa kunci API, atau kombinasi nama pengguna dan kata sandi. Ini berarti Anda juga perlu membuat peran IAM yang memungkinkan akses OpenSearch Layanan untuk membaca dari Secrets Manager.

Topik

Prasyarat

Untuk membuat konektor untuk Cohere atau penyedia eksternal apa pun dengan OpenSearch Layanan, Anda harus memiliki peran IAM yang memberikan akses OpenSearch Layanan AWS Secrets Manager, tempat Anda menyimpan kredensialnya. Anda juga harus menyimpan kredensialnya di Secrets Manager.

Membuat peran IAM

Siapkan peran IAM untuk mendelegasikan izin Secrets Manager ke Layanan. OpenSearch Anda juga dapat menggunakan `SecretManagerReadWrite` peran yang ada. Untuk membuat peran baru, lihat Membuat peran IAM (konsol) di Panduan Pengguna IAM. Jika Anda membuat peran baru alih-alih menggunakan peran AWS terkelola, ganti `opensearch-secretmanager-role` dalam tutorial ini dengan nama peran Anda sendiri.

1. Lampirkan kebijakan IAM terkelola berikut ke peran baru Anda untuk memungkinkan OpenSearch Layanan mengakses nilai Secrets Manager Anda. Untuk melampirkan kebijakan ke peran, lihat Menambahkan Izin Identitas IAM.

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

2. Ikuti instruksi dalam Memodifikasi kebijakan kepercayaan peran untuk mengedit hubungan kepercayaan peran. Anda harus menentukan OpenSearch Layanan dalam `Principal` pernyataan:

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

Kami menyarankan Anda menggunakan tombol `aws:SourceAccount` dan `aws:SourceArn` kondisi untuk membatasi akses ke domain tertentu. Itu `SourceAccount` adalah Akun AWS

ID milik pemilik domain, dan `SourceArn` adalah ARN dari domain. Misalnya, Anda dapat menambahkan blok kondisi berikut ke kebijakan kepercayaan:

```
"Condition": {
    "StringEquals": {
        "aws:SourceAccount": "account-id"
    },
    "ArnLike": {
        "aws:SourceArn": "arn:aws:es:region:account-id:domain/domain-name"
    }
}
```

Konfigurasi izin

Untuk membuat konektor, Anda memerlukan izin untuk meneruskan peran IAM ke OpenSearch Layanan. Anda juga memerlukan akses ke tindakan `es:ESHttpPost`. Untuk memberikan kedua izin ini, lampirkan kebijakan berikut ke peran IAM yang kredensialnya digunakan untuk menandatangani permintaan:

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Effect": "Allow",
      "Action": "iam:PassRole",
      "Resource": "arn:aws:iam::account-id:role/opensearch-secretmanager-role"
    },
    {
      "Effect": "Allow",
      "Action": "es:ESHttpPost",
      "Resource": "arn:aws:es:region:account-id:domain/domain-name/*"
    }
  ]
}
```

Jika pengguna atau peran Anda tidak memiliki `iam:PassRole` izin untuk meneruskan peran Anda, Anda mungkin mengalami kesalahan otorisasi saat mencoba mendaftarkan repositori di langkah berikutnya.

Mengatur AWS Secrets Manager

Untuk menyimpan kredensi otorisasi Anda di Secrets Manager, lihat Membuat AWS Secrets Manager rahasia di Panduan Pengguna.AWS Secrets Manager

Setelah Secrets Manager menerima pasangan nilai kunci Anda sebagai rahasia, Anda menerima ARN dengan format:. `arn:aws:secretsmanager:us-west-2:123456789012:secret:MySecret-a1b2c3` Simpan catatan ARN ini, saat Anda menggunakannya dan kunci Anda saat membuat konektor di langkah berikutnya.

Petakan peran ML di OpenSearch Dasbor (jika menggunakan kontrol akses berbutir halus)

Kontrol akses berbutir halus memperkenalkan langkah tambahan saat menyiapkan konektor. Bahkan jika Anda menggunakan autentikasi basic HTTP untuk semua tujuan lain, Anda perlu memetakan peran `ml_full_access` ke IAM role Anda yang memiliki izin `iam:PassRole` untuk meneruskan `opensearch-sagemaker-role`.

1. Arahkan ke plugin OpenSearch Dasbor untuk domain OpenSearch Layanan Anda. Anda dapat menemukan titik akhir Dasbor di dasbor domain Anda di konsol OpenSearch Layanan.
2. Dari menu utama pilih Keamanan, Peran, dan pilih peran ml_full_access.
3. Pilih Pengguna yang Dipetakan, Kelola pemetaan.
4. Di bawah peran Backend, tambahkan ARN dari peran yang memiliki izin untuk diteruskan. `opensearch-sagemaker-role`

   ```
   arn:aws:iam::account-id:role/role-name
   ```

5. Pilih Peta dan konfirmasikan pengguna atau peran muncul di bawah Pengguna yang dipetakan.

Buat konektor OpenSearch Service

Untuk membuat konektor, kirim `POST` permintaan ke titik akhir domain OpenSearch Layanan. Anda dapat menggunakan curl, contoh klien Python, Postman, atau metode lain untuk mengirim permintaan yang ditandatangani. Perhatikan bahwa Anda tidak dapat menggunakan `POST` permintaan di konsol Kibana. Permintaan mengambil format berikut:

```
POST domain-endpoint/_plugins/_ml/connectors/_create
{
```

```
    "name": "Cohere Connector: embedding",
    "description": "The connector to cohere embedding model",
    "version": 1,
    "protocol": "http",
    "credential": {
        "secretArn": "arn:aws:secretsmanager:region:account-id:secret:cohere-key-id",
        "roleArn": "arn:aws:iam::account-id:role/opensearch-secretmanager-role"
    },
    "actions": [
        {
            "action_type": "predict",
            "method": "POST",
            "url": "https://api.cohere.ai/v1/embed",
            "headers": {
                "Authorization": "Bearer ${credential.secretArn.cohere-key-used-in-
secrets-manager}"
            },
            "request_body": "{ \"texts\": ${parameters.texts}, \"truncate\": \"END\" }"
        }
    ]
}
```

Badan permintaan untuk permintaan ini berbeda dari permintaan konektor sumber terbuka dalam dua cara. Di dalam `credential` lapangan, Anda melewati ARN untuk peran IAM yang memungkinkan OpenSearch Layanan membaca dari Secrets Manager, bersama dengan ARN untuk rahasia apa. Di `headers` lapangan, Anda merujuk pada rahasia menggunakan kunci rahasia dan fakta itu berasal dari ARN.

Jika domain Anda berada dalam virtual private cloud (VPC), komputer Anda harus terhubung ke VPC agar permintaan berhasil membuat AI connetor. Mengakses VPC bervariasi menurut konfigurasi jaringan, tetapi biasanya melibatkan koneksi ke VPN atau jaringan perusahaan. Untuk memastikan bahwa Anda dapat mencapai domain OpenSearch Layanan, `https://your-vpc-domain.region.es.amazonaws.com` navigasikan ke browser web dan verifikasi bahwa Anda menerima respons JSON default.

Contoh klien Python

Klien Python lebih sederhana untuk diotomatisasi daripada permintaan HTTP dan memiliki kegunaan ulang yang lebih baik. Untuk membuat konektor AI dengan klien Python, simpan kode contoh berikut ke file Python. Klien membutuhkan AWS SDK for Python (Boto3), `requests`, dan `requests-aws4auth`paket.

```
import boto3
import requests
from requests_aws4auth import AWS4Auth

host = 'domain-endpoint/'
region = 'region'
service = 'es'
credentials = boto3.Session().get_credentials()
awsauth = AWS4Auth(credentials.access_key, credentials.secret_key, region, service,
 session_token=credentials.token)

path = '_plugins/_ml/connectors/_create'
url = host + path

payload = {
    "name": "Cohere Connector: embedding",
    "description": "The connector to cohere embedding model",
    "version": 1,
    "protocol": "http",
    "credential": {
        "secretArn": "arn:aws:secretsmanager:region:account-id:secret:cohere-key-id",
        "roleArn": "arn:aws:iam::account-id:role/opensearch-secretmanager-role"
    },
    "actions": [
        {
            "action_type": "predict",
            "method": "POST",
            "url": "https://api.cohere.ai/v1/embed",
            "headers": {
                "Authorization": "Bearer ${credential.secretArn.cohere-key-used-in-
secrets-manager}"
            },
            "request_body": "{ \"texts\": ${parameters.texts}, \"truncate\": \"END\" }"
        }
    ]
}

headers = {"Content-Type": "application/json"}

r = requests.post(url, auth=awsauth, json=payload, headers=headers)
print(r.status_code)
print(r.text)
```

Menggunakan AWS CloudFormation untuk mengatur inferensi jarak jauh untuk pencarian semantik

Dimulai dengan OpenSearch versi 2.9, Anda dapat menggunakan inferensi jarak jauh dengan pencarian semantik untuk meng-host model machine learning (ML) Anda sendiri. Inferensi jarak jauh menggunakan plugin ML Commons untuk memungkinkan Anda meng-host inferensi model Anda dari jarak jauh pada layanan ML, seperti dan Amazon SageMaker Amazon BedRock, dan menghubungkannya ke OpenSearch Layanan Amazon dengan konektor ML.

Untuk memudahkan penyiapan inferensi jarak jauh, Amazon OpenSearch Service menyediakan AWS CloudFormationtemplate di konsol. CloudFormation adalah sebuah Layanan AWS yang memungkinkan Anda memodelkan, menyediakan, dan mengelola AWS dan sumber daya pihak ketiga dengan memperlakukan infrastruktur sebagai kode.

OpenSearch CloudFormation Template mengotomatiskan proses penyediaan model untuk Anda, sehingga Anda dapat dengan mudah membuat model di domain OpenSearch Layanan Anda dan kemudian menggunakan ID model untuk menelan data dan menjalankan kueri penelusuran saraf.

Topik

Prasyarat

Untuk menggunakan CloudFormation template dengan OpenSearch Service, lengkapi prasyarat berikut.

Menyiapkan domain OpenSearch Layanan

Sebelum dapat menggunakan CloudFormation templat, Anda harus menyiapkan domain OpenSearch Layanan Amazon dengan versi 2.9 atau yang lebih baru dan kontrol akses berbutir halus diaktifkan. Buat peran backend OpenSearch Layanan untuk memberikan izin plugin MLCommons untuk membuat konektor untuk Anda.

CloudFormation Template membuat peran Lambda IAM untuk Anda dengan nama default`LambdaInvokeOpenSearchMLCommonsRole`, yang dapat Anda ganti jika Anda ingin memilih nama yang berbeda. Setelah template membuat peran IAM ini, Anda perlu memberikan izin

fungsi Lambda untuk memanggil domain Layanan OpenSearch Anda. Untuk melakukannya, petakan peran yang diberi nama `ml_full_access` ke peran backend OpenSearch Service Anda dengan langkah-langkah berikut:

1. Arahkan ke plugin OpenSearch Dasbor untuk domain OpenSearch Layanan Anda. Anda dapat menemukan titik akhir Dasbor di dasbor domain Anda di konsol OpenSearch Layanan.
2. Dari menu utama pilih Keamanan, Peran, dan pilih peran ml_full_access.
3. Pilih Pengguna yang Dipetakan, Kelola pemetaan.
4. Di bawah peran Backend, tambahkan ARN dari peran Lambda yang memerlukan izin untuk memanggil domain Anda.

   ```
   arn:aws:iam::account-id:role/role-name
   ```

5. Pilih Peta dan konfirmasikan pengguna atau peran muncul di bawah Pengguna yang dipetakan.

Setelah Anda memetakan peran, navigasikan ke konfigurasi keamanan domain Anda dan tambahkan peran Lambda IAM ke kebijakan akses Layanan OpenSearch Anda.

Aktifkan izin pada Akun AWS

Anda Akun AWS harus memiliki izin untuk mengakses CloudFormation dan Lambda, bersama dengan mana pun yang Layanan AWS Anda pilih untuk template Anda — baik Runtime atau Amazon. SageMaker BedRock

Jika Anda menggunakan Amazon Bedrock, Anda juga harus mendaftarkan model Anda. Lihat Akses model di Panduan Pengguna Amazon Bedrock untuk mendaftarkan model Anda.

Jika Anda menggunakan bucket Amazon S3 Anda sendiri untuk menyediakan artefak model, Anda harus menambahkan peran CloudFormation IAM ke kebijakan akses S3 Anda. Untuk informasi lebih lanjut, lihat Menambahkan dan menghapus izin identitas IAM dalam Panduan Pengguna IAM.

Amazon SageMaker template

SageMaker CloudFormation Templat Amazon menentukan beberapa AWS sumber daya untuk menyiapkan plugin saraf dan pencarian semantik untuk Anda.

Pertama, gunakan Integrasi dengan model penyematan teks melalui SageMaker template Amazon untuk menerapkan model penyematan teks di SageMaker Runtime sebagai server. Jika Anda tidak menyediakan titik akhir model, CloudFormation buat peran IAM yang memungkinkan

SageMaker Runtime mengunduh artefak model dari Amazon S3 dan menerapkannya ke server. Jika Anda memberikan titik akhir, CloudFormation buat peran IAM yang memungkinkan fungsi Lambda mengakses domain OpenSearch Layanan atau, jika peran sudah ada, memperbarui dan menggunakan kembali peran tersebut. Endpoint melayani model jarak jauh yang digunakan untuk konektor ML dengan plugin MLCommons.

Selanjutnya, gunakan template Integrasi dengan Sparse Encoders melalui Amazon Sagemaker untuk membuat fungsi Lambda yang membuat domain Anda menyiapkan konektor inferensi jarak jauh. Setelah konektor dibuat di OpenSearch Service, inferensi jarak jauh dapat menjalankan pencarian semantik menggunakan model jarak jauh di Runtime. SageMaker Template mengembalikan ID model di domain Anda kembali kepada Anda sehingga Anda dapat mulai mencari.

Untuk menggunakan SageMaker CloudFormation template Amazon

1. Buka konsol OpenSearch Layanan Amazon di https://console.aws.amazon.com/aos/home.
2. Di navigasi kiri, pilih Integrasi.
3. Di bawah setiap SageMaker templat Amazon, pilih Konfigurasi domain, Konfigurasikan domain publik.
4. Ikuti prompt di CloudFormation konsol untuk menyediakan tumpukan Anda dan menyiapkan model.

> **Note**
>
> OpenSearch Layanan juga menyediakan template terpisah untuk mengkonfigurasi domain VPC. Jika Anda menggunakan template ini, Anda perlu memberikan ID VPC untuk fungsi Lambda.

Templat Batuan Dasar Amazon

Mirip dengan SageMaker CloudFormation template Amazon, template Amazon Bedrock CloudFormation menyediakan AWS sumber daya yang diperlukan untuk membuat konektor antara OpenSearch Service dan Amazon Bedrock.

Pertama, template membuat peran IAM yang memungkinkan fungsi Lambda future mengakses domain Layanan OpenSearch Anda. Template kemudian membuat fungsi Lambda, yang memiliki domain membuat konektor menggunakan plugin MLCommons. Setelah OpenSearch Service

membuat konektor, penyiapan inferensi jarak jauh selesai dan Anda dapat menjalankan pencarian semantik menggunakan operasi Amazon Bedrock API.

Perhatikan bahwa karena Amazon Bedrock meng-host model ML-nya sendiri, Anda tidak perlu menerapkan model ke SageMaker Runtime. Sebagai gantinya, template menggunakan titik akhir yang telah ditentukan untuk Amazon Bedrock dan melewatkan langkah penyediaan titik akhir.

Untuk menggunakan template Amazon Bedrock CloudFormation

1. Buka konsol OpenSearch Layanan Amazon di https://console.aws.amazon.com/aos/home.
2. Di navigasi kiri, pilih Integrasi.
3. Di bawah Integrasikan dengan Amazon Titan Text Embeddings model melalui Amazon Bedrock, pilih Konfigurasi domain, Konfigurasikan domain publik.
4. Ikuti prompt untuk mengatur model Anda.

> **Note**
>
> OpenSearch Layanan juga menyediakan template terpisah untuk mengkonfigurasi domain VPC. Jika Anda menggunakan template ini, Anda perlu memberikan ID VPC untuk fungsi Lambda.

Selain itu, OpenSearch Service menyediakan template Amazon Bedrock berikut untuk terhubung ke model Cohere dan model penyematan multimodal Amazon Titan:

- `Integration with Cohere Embed through Amazon Bedrock`
- `Integrate with Amazon Bedrock Titan Multi-modal`

Pengaturan ML Commons yang tidak didukung

Amazon OpenSearch Service tidak mendukung penggunaan setelan ML Commons berikut:

- `plugins.ml_commons.allow_registering_model_via_url`
- `plugins.ml_commons.allow_registering_model_via_local_file`

Untuk informasi selengkapnya tentang setelan ML Commons, lihat setelan klaster MLCommons.

Analisis Keamanan untuk OpenSearch Layanan Amazon

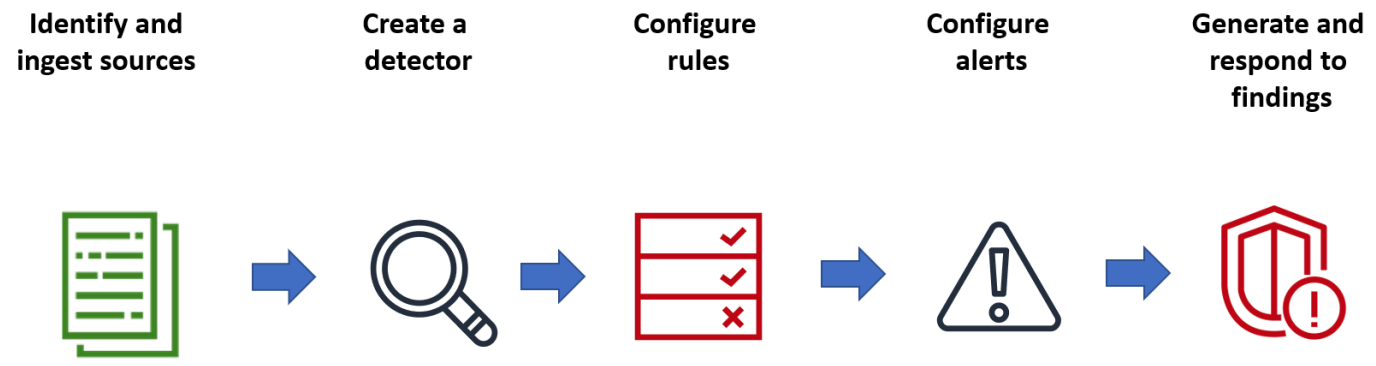
Security Analytics adalah OpenSearch solusi yang memberikan visibilitas ke infrastruktur organisasi Anda, memantau aktivitas anomali, mendeteksi potensi ancaman keamanan secara real time, dan memicu peringatan ke tujuan yang telah dikonfigurasi sebelumnya. Anda dapat memantau aktivitas berbahaya dari log peristiwa keamanan Anda dengan terus mengevaluasi aturan keamanan dan meninjau temuan keamanan yang dibuat secara otomatis. Selain itu, Security Analytics dapat menghasilkan peringatan otomatis dan mengirimkannya ke saluran notifikasi tertentu, seperti Slack atau email.

Anda dapat menggunakan plugin Security Analytics untuk mendeteksi ancaman umum out-of-the-box dan menghasilkan wawasan keamanan penting dari log peristiwa keamanan yang ada, seperti log firewall, log windows, dan log audit otentikasi. Untuk menggunakan Security Analytics, domain Anda harus menjalankan OpenSearch versi 2.5 atau yang lebih baru.

Untuk informasi selengkapnya tentang mengonfigurasi plugin Analytics Keamanan, lihat Analisis Keamanan di OpenSearch dokumentasi.

Komponen dan konsep analitik keamanan

Sejumlah alat dan fitur memberikan dasar bagi pengoperasian Analisis Keamanan. Komponen utama yang menyusun plugin termasuk detektor, jenis log, aturan, temuan, dan peringatan.



Jenis log

OpenSearch mendukung beberapa jenis log dan menyediakan out-of-the-box pemetaan untuk setiap jenis. Anda menentukan jenis log dan mengonfigurasi interval waktu saat membuat detektor, dan

dari sana Security Analytics secara otomatis mengaktifkan seperangkat aturan yang relevan yang berjalan pada interval tersebut.

Detektor

Detektor mengidentifikasi berbagai ancaman keamanan siber untuk jenis log di seluruh indeks data Anda. Anda mengonfigurasi detektor untuk menggunakan aturan kustom dan aturan Sigma pra-paket yang mengevaluasi peristiwa yang terjadi di sistem. Detektor kemudian menghasilkan temuan keamanan dari peristiwa ini. Untuk informasi selengkapnya tentang detektor, lihat Membuat detektor dalam dokumentasi. OpenSearch

Aturan

Aturan deteksi ancaman menentukan kondisi yang diterapkan detektor pada data log yang dicerna untuk mengidentifikasi peristiwa keamanan. Security Analytics mendukung pengimporan, pembuatan, dan penyesuaian aturan untuk memenuhi kebutuhan Anda, dan juga menyediakan aturan Sigma sumber terbuka yang dikemas untuk mendeteksi ancaman umum dari log Anda. Security Analytics memetakan banyak aturan ke basis pengetahuan yang terus berkembang tentang taktik dan teknik musuh yang dikelola oleh organisasi MITRE ATT&CK. Anda dapat menggunakan OpenSearch Dasbor atau API untuk membuat dan menggunakan aturan. Untuk informasi selengkapnya tentang aturan, lihat Bekerja dengan aturan dalam OpenSearch dokumentasi.

Temuan

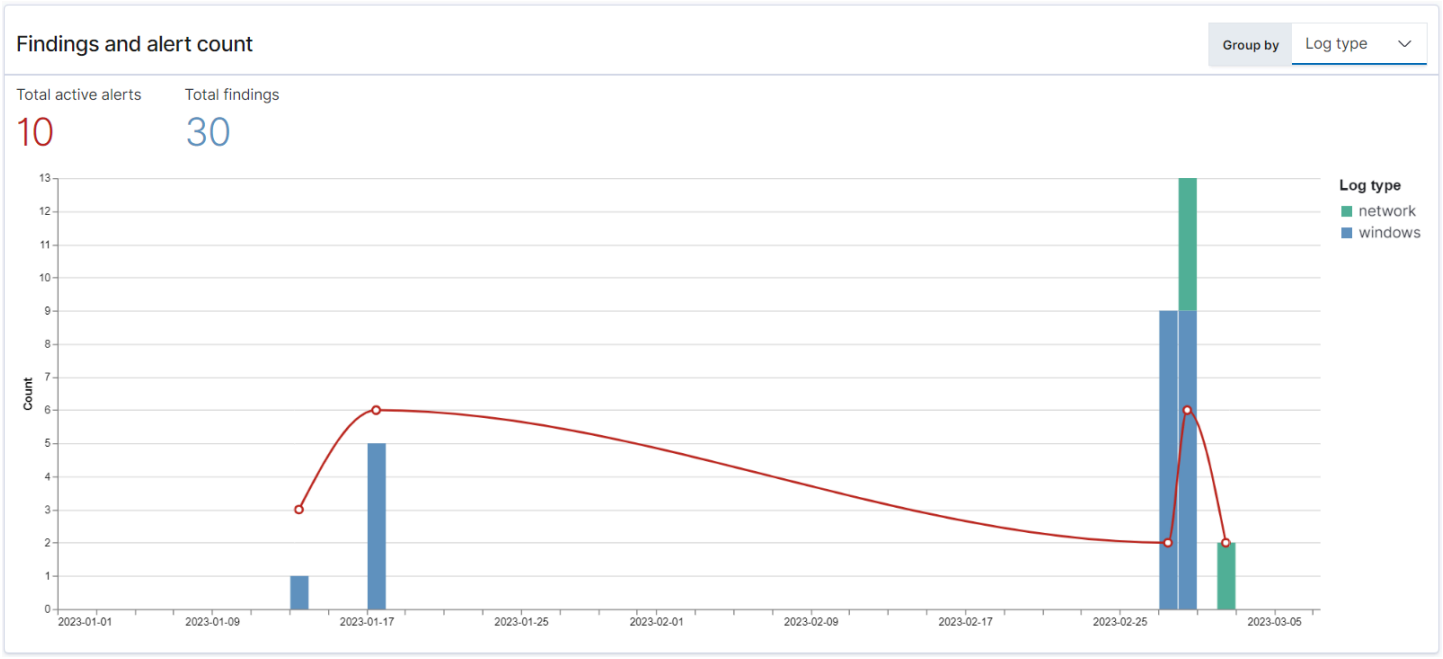
Ketika detektor mencocokkan aturan dengan peristiwa log, detektor menghasilkan temuan. Setiap temuan mencakup kombinasi unik dari aturan pilih, jenis log, dan tingkat keparahan aturan. Temuan tidak selalu menunjukkan ancaman yang akan segera terjadi dalam sistem, tetapi mereka selalu mengisolasi peristiwa yang menarik. Untuk informasi lebih lanjut tentang temuan, lihat Bekerja dengan temuan dalam OpenSearch dokumentasi.

Peringatan

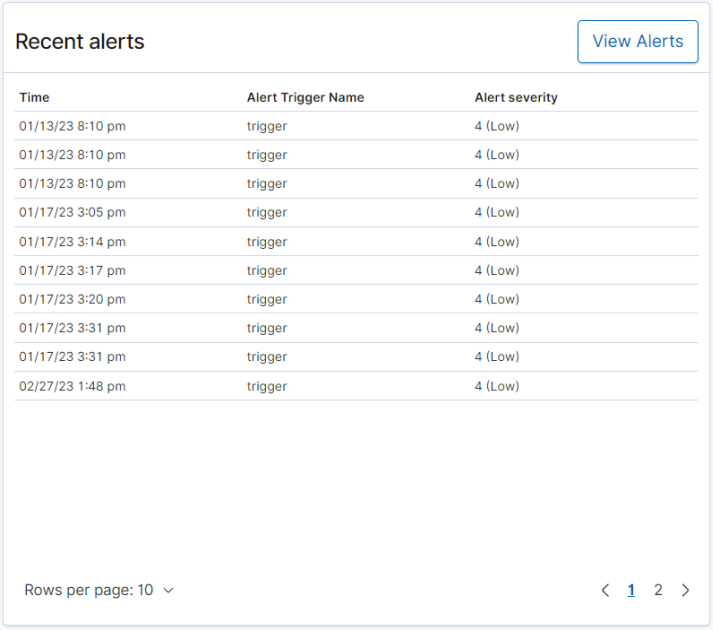
Saat membuat detektor, Anda dapat menentukan satu atau beberapa kondisi yang memicu peringatan. Peringatan adalah pemberitahuan yang dikirim ke saluran pilihan, seperti Slack atau email. Anda menyetel peringatan yang akan dipicu saat detektor cocok dengan satu atau beberapa aturan, dan dapat menyesuaikan pesan notifikasi. Untuk informasi selengkapnya tentang lansiran, lihat Bekerja dengan lansiran di dokumentasi. OpenSearch

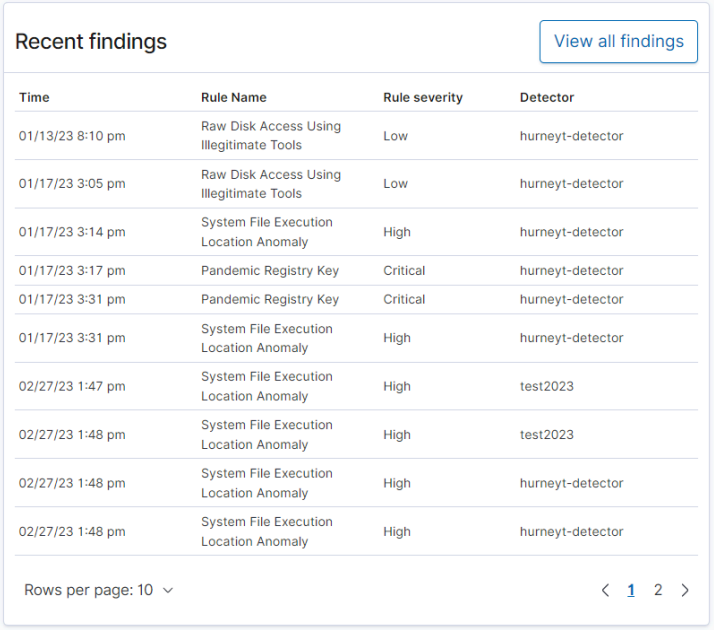
Menjelajahi Analisis Keamanan

Anda dapat menggunakan OpenSearch Dasbor untuk memvisualisasikan dan mendapatkan wawasan tentang plugin Analisis Keamanan Anda. Tampilan Ikhtisar memberikan informasi seperti temuan dan jumlah peringatan, temuan dan peringatan terbaru, aturan deteksi yang sering, dan daftar detektor Anda. Anda dapat melihat tampilan ringkasan yang terdiri dari beberapa visualisasi. Bagan berikut, misalnya, menunjukkan tren temuan dan peringatan untuk berbagai jenis log selama periode waktu tertentu.

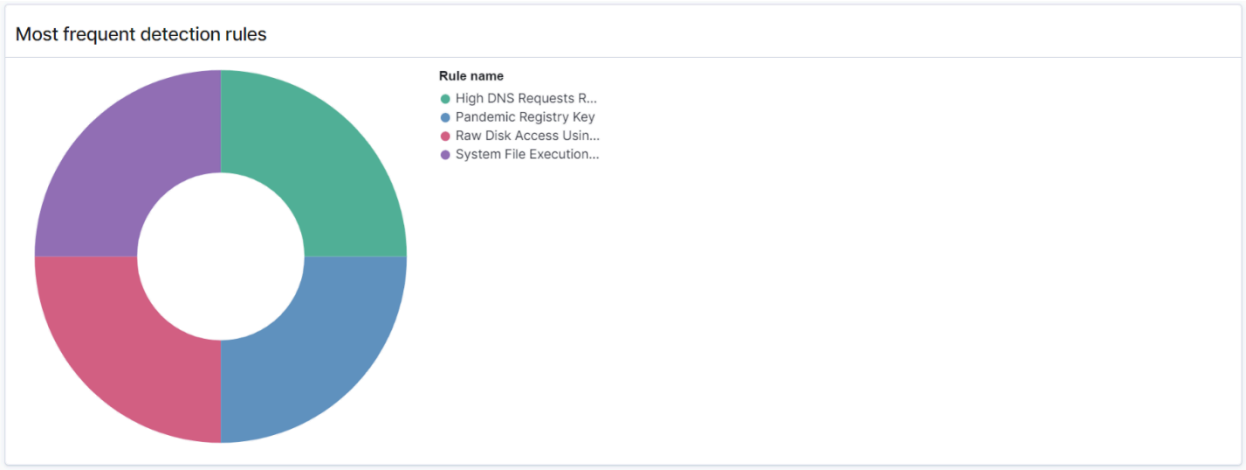


Lebih jauh ke bawah halaman, Anda dapat meninjau temuan dan peringatan terbaru Anda.





Selain itu, Anda dapat melihat distribusi aturan yang paling sering dipicu di semua detektor aktif. Ini dapat membantu Anda mendeteksi dan menyelidiki berbagai jenis aktivitas berbahaya di seluruh jenis log.



Akhirnya, Anda dapat melihat status detektor yang dikonfigurasi. Dari panel ini, Anda juga dapat menavigasi ke alur kerja create detector.

Untuk mengonfigurasi penyiapan Analisis Keamanan Anda, buat aturan dengan halaman Aturan dan gunakan aturan tersebut untuk menulis detektor di halaman Detektor. Untuk tampilan hasil Analisis Keamanan yang lebih terfokus, Anda dapat menggunakan halaman Temuan dan Peringatan.

Konfigurasi izin

Jika Anda mengaktifkan Analisis Keamanan pada domain OpenSearch Layanan yang sudah ada sebelumnya, `security_analytics_manager` peran tersebut mungkin tidak ditentukan pada domain. Pengguna non-admin harus dipetakan ke peran ini untuk mengelola indeks hangat pada domain menggunakan kontrol akses berbutir halus. Untuk membuat secara manual peran `security_analytics_manager`, lakukan langkah-langkah berikut:

1. Di OpenSearch Dasbor, buka Keamanan dan pilih Izin.
2. Pilih Buat grup tindakan dan konfigurasi grup-grup berikut:

Nama grup	Izin
`security_analytics_full_access`	• `cluster:admin/opensearch/securityanalytics/alerts/*` • `cluster:admin/opensearch/securityanalytics/detector/*` • `cluster:admin/opensearch/securityanalytics/findings/*` • `cluster:admin/opensearch/securityanalytics/mapping/*` • `cluster:admin/opensearch/securityanalytics/rule/*`

Nama grup	Izin
security_analytics_read_access	• cluster:admin/opensearch/securityanalytics/alerts/get • cluster:admin/opensearch/securityanalytics/detector/get • cluster:admin/opensearch/securityanalytics/detector/search • cluster:admin/opensearch/securityanalytics/findings/get • cluster:admin/opensearch/securityanalytics/mapping/get • cluster:admin/opensearch/securityanalytics/mapping/view/get • cluster:admin/opensearch/securityanalytics/rule/get • cluster:admin/opensearch/securityanalytics/rule/search

3. Pilih Peran dan Buat peran.
4. Beri nama peran security_analytics_manager.
5. Untuk Izin klaster, pilih `security_analytics_full_access` dan `security_analytics_read_access`.
6. Untuk Indeks, ketik *.
7. Untuk izin Indeks, pilih `indices:admin/mapping/put` dan`indices:admin/mappings/get`.
8. Pilih Buat.
9. Setelah Anda membuat peran, petakan ke setiap pengguna atau peran backend yang akan mengelola indeks Analisis Keamanan.

Memecahkan masalah

Tidak ada kesalahan indeks seperti itu

Jika Anda tidak memiliki detektor dan Anda membuka dasbor Analytics Keamanan, Anda mungkin melihat pemberitahuan di kanan bawah yang bertuliskan`[index_not_found_exception] no such index [.opensearch-sap-detectors-config]`. Anda dapat mengabaikan pemberitahuan ini, yang menghilang dalam beberapa detik dan tidak akan muncul lagi setelah Anda membuat detektor.

Observabilitas di Amazon OpenSearch Layanan

Instalasi default OpenSearch Dasbor untuk Amazon OpenSearch Layanan mencakup plugin Observability, yang dapat Anda gunakan untuk memvisualisasikan peristiwa berbasis data menggunakan Piped Processing Language (PPL) untuk menjelajahi, menemukan, dan data kueri yang disimpan di OpenSearch. Plugin membutuhkan OpenSearch 1.2 atau versi yang lebih baru.

Plugin Observability menyediakan pengalaman terpadu untuk mengumpulkan dan memantau metrik, log, dan jejak dari sumber data umum. Pengumpulan dan pemantauan data di satu tempat memungkinkan penumpukan penuh, end-to-end pengamatan seluruh infrastruktur Anda. Dokumentasi lengkap untuk plugin Observability ada di[Dokumentasi OpenSearch](#).

Proses setiap orang untuk menjelajahi data berbeda. Jika Anda baru mengeksplorasi data dan membuat visualisasi, sebaiknya coba alur kerja seperti berikut ini:

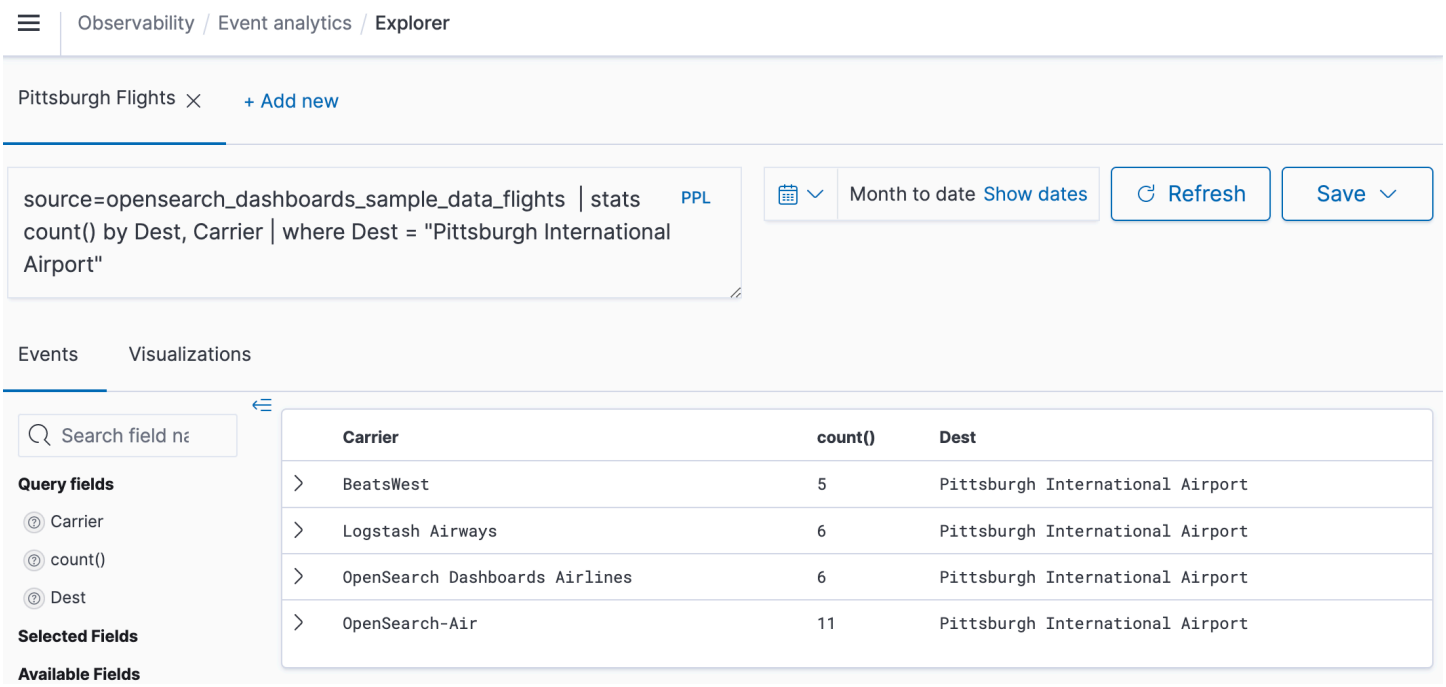
Jelajahi data Anda dengan analitik peristiwa

Untuk memulai, katakanlah bahwa Anda mengumpulkan data penerbangan di OpenSearch Domain layanan dan Anda ingin mengetahui maskapai penerbangan mana yang paling banyak tiba di Bandara Internasional Pittsburgh bulan lalu. Anda menulis query PPL berikut:

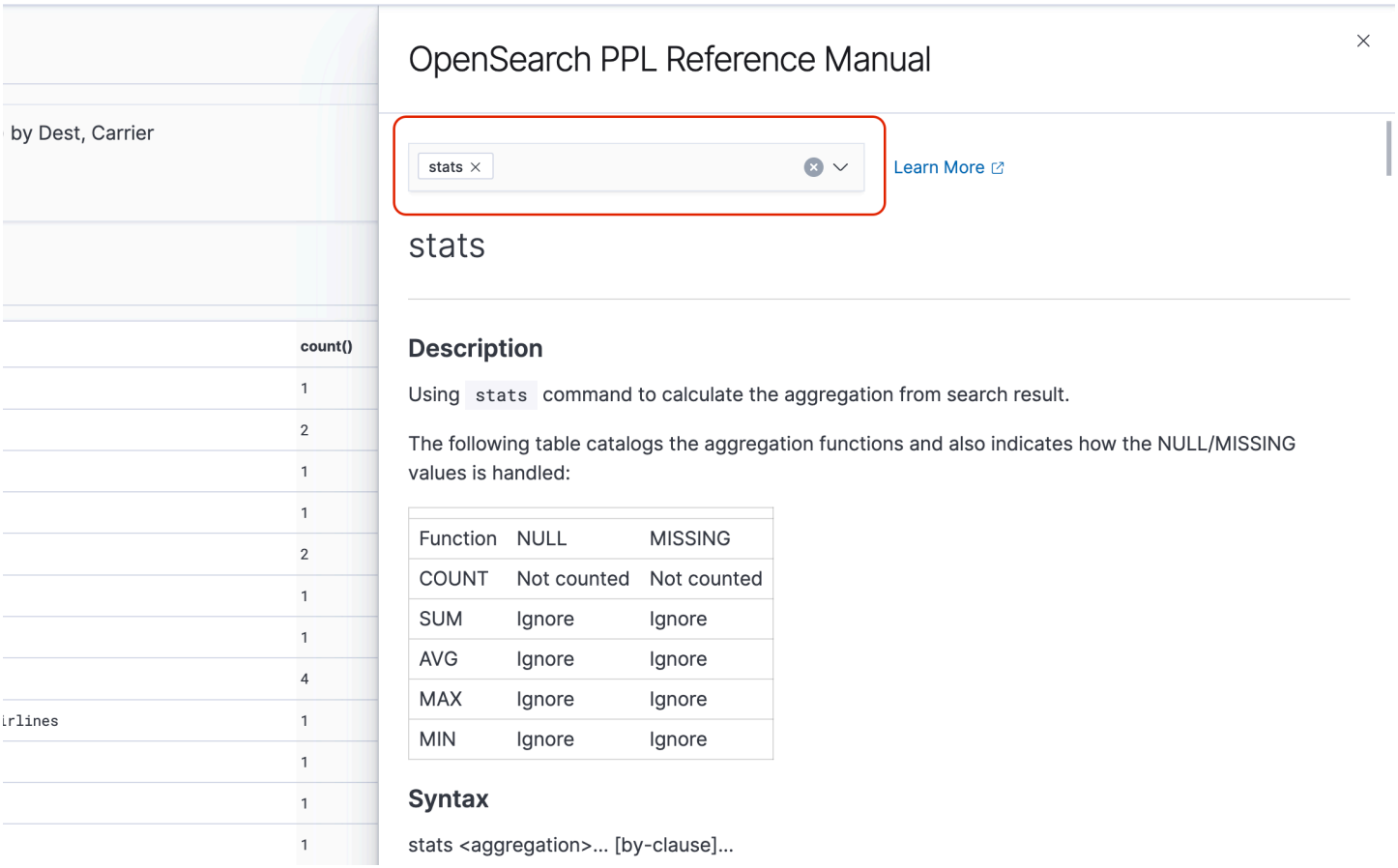
```
source=opensearch_dashboards_sample_data_flights |
    stats count() by Dest, Carrier |
    where Dest = "Pittsburgh International Airport"
```

Query ini menarik data dari indeks bernama`opensearch_dashboards_sample_data_flights`. Kemudian menggunakan`stats`perintah untuk mendapatkan jumlah total penerbangan dan kelompok itu sesuai dengan bandara tujuan dan operator. Akhirnya, ia menggunakan`where`klausul untuk menyaring hasil penerbangan yang tiba di Bandara Internasional Pittsburgh.

Berikut adalah apa data terlihat seperti ketika ditampilkan selama bulan lalu:



Anda dapat memilihPPLtombol di editor query untuk mendapatkan informasi penggunaan dan contoh untuk setiap perintah PPL:



Mari kita lihat contoh yang lebih kompleks, yang meminta informasi tentang penundaan penerbangan:

```
source=opensearch_dashboards_sample_data_flights |
    where FlightDelayMin > 0 |
    stats sum(FlightDelayMin) as minimum_delay, count() as total_delayed by Carrier,
 Dest |
    eval avg_delay=minimum_delay / total_delayed |
    sort - avg_delay
```

Setiap perintah dalam kueri memengaruhi output akhir:

- `source=opensearch_dashboards_sample_data_flights`- menarik data dari indeks yang sama seperti contoh sebelumnya
- `where FlightDelayMin > 0`- menyaring data ke penerbangan yang tertunda
- `stats sum(FlightDelayMin) as minimum_delay, count() as total_delayed by Carrier`- untuk setiap operator, mendapatkan total waktu tunda minimum dan jumlah total penerbangan tertunda
- `eval avg_delay=minimum_delay / total_delayed`- menghitung waktu tunda rata-rata untuk setiap operator dengan membagi waktu tunda minimum dengan jumlah total penerbangan yang tertunda
- `sort - avg_delay`- mengurutkan hasil dengan keterlambatan rata-rata dalam urutan menurun

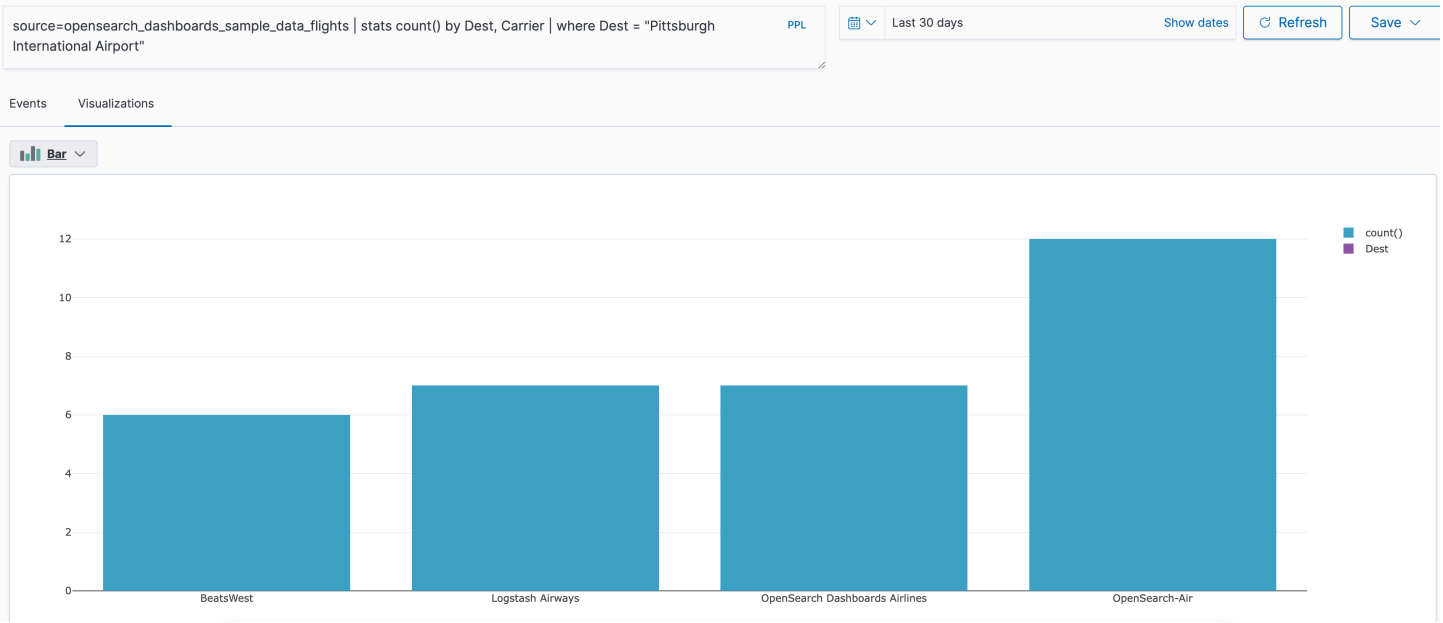
Dengan kueri ini, Anda dapat menentukannya OpenSearch Dasbor Airlines memiliki, rata-rata, lebih sedikit penundaan.

	avg_delay	Carrier	minimum_delay	total_delayed
>	212	Logstash Airways	4470	21
>	184	OpenSearch-Air	4245	23
>	155	BeatsWest	2025	13
>	153	OpenSearch Dashboards Airlines	4305	28

Anda dapat menemukan lebih banyak contoh permintaan PPL di bawahPertanyaan dan VisualisasipadaAnalitik acarahalaman.

Buat visualisasi

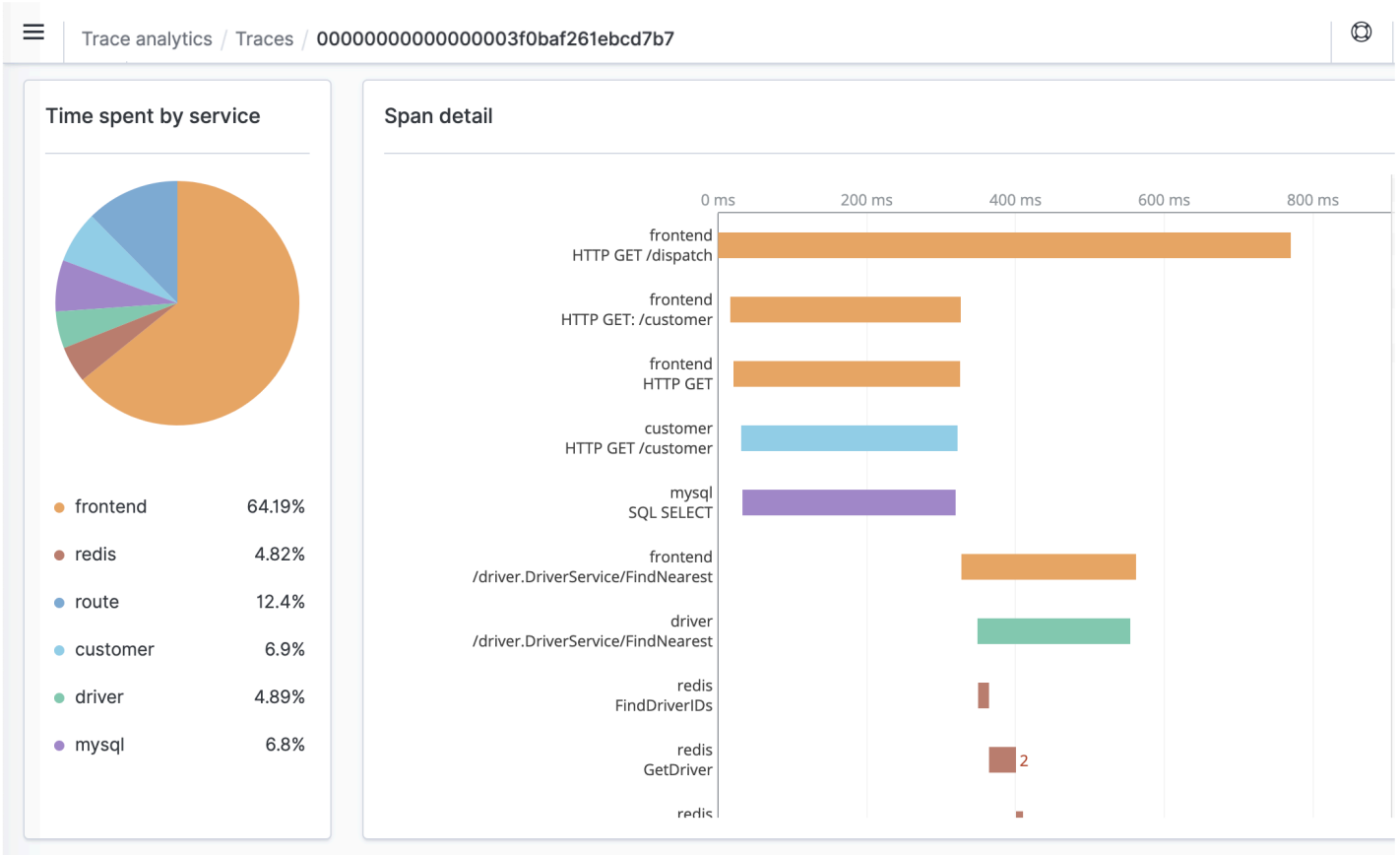
Setelah Anda mengkueri data yang Anda minati dengan benar, Anda dapat menyimpan kueri tersebut sebagai visualisasi:



Kemudian tambahkan visualisasi tersebut kepanel operasionaluntuk membandingkan potongan data yang berbeda. Leveragebuku catatanuntuk menggabungkan visualisasi dan blok kode yang berbeda yang dapat Anda bagikan dengan anggota tim.

Menyelam lebih dalam dengan Trace Analytics

Trace Analyticsmenyediakan cara untuk memvisualisasikan aliran peristiwa di OpenSearch data untuk mengidentifikasi dan memperbaiki masalah kinerja dalam aplikasi terdistribusi.

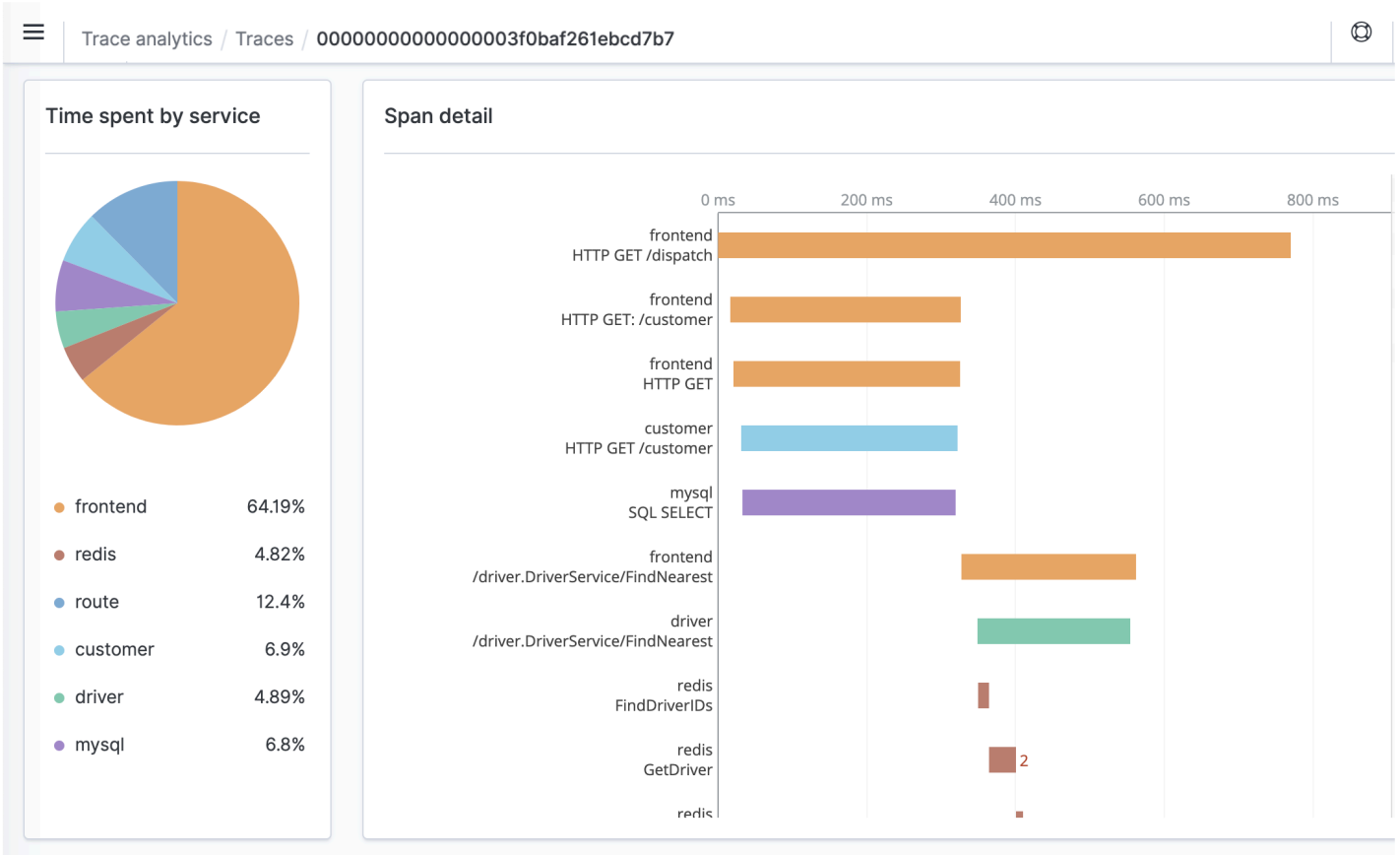


Trace Analytics untuk Amazon OpenSearch Service

Anda dapat menggunakan Trace Analytics, yang merupakan bagian dari plugin OpenSearch Observability, untuk menganalisis data pelacakan dari aplikasi terdistribusi. Trace Analytics membutuhkan OpenSearch atau Elasticsearch 7.9 atau yang lebih baru.

Dalam aplikasi terdistribusi, operasi tunggal, seperti pengguna mengklik tombol, dapat memicu serangkaian peristiwa yang luas. Sebagai contoh, front end aplikasi mungkin memanggil layanan backend, yang memanggil layanan lain, yang mengkueri basis data, yang memproses kueri dan mengembalikan hasil. Kemudian layanan backend pertama mengirimkan konfirmasi ke front end, yang memperbarui UI.

Anda dapat menggunakan Trace Analytics untuk membantu Anda memvisualisasikan alur peristiwa ini dan mengidentifikasi masalah performa.



Prasyarat

Trace Analytics mengharuskan Anda untuk menambahkan instrumentasi ke aplikasi Anda dan menghasilkan data pelacakan menggunakan pustaka yang OpenTelemetry didukung seperti Jaeger atau Zipkin. Langkah ini terjadi sepenuhnya di luar OpenSearch Service. AWSDistro untuk OpenTelemetry dokumentasi berisi contoh aplikasi untuk banyak bahasa pemrograman yang dapat membantu Anda memulai, termasuk Java, Python, Go, dan. JavaScript

Setelah Anda menambahkan instrumentasi ke aplikasi Anda, OpenTelemetryCollector menerima data dari aplikasi dan memformatnya menjadi OpenTelemetry data. Lihat daftar penerima di GitHub. AWS Distro untuk OpenTelemetry termasuk penerima untuk AWS X-Ray.

Akhirnya, Data Prepper, OpenSearch komponen independen, memformat OpenTelemetry data untuk digunakan. OpenSearch Data Prepper berjalan pada mesin di luar klaster OpenSearch Service, mirip dengan Logstash.

Untuk file Docker Compose yang menunjukkan aliran data end-to-end, lihat dokumentasi. OpenSearch

OpenTelemetryKonfigurasi sampel kolektor

Untuk menggunakan OpenTelemetry Collector dengan Amazon OpenSearch Ingestion, cobalah konfigurasi sampel berikut:

```
extensions:
  sigv4auth:
    region: "us-east-1"
    service: "osis"

receivers:
  jaeger:
    protocols:
      grpc:

exporters:
  otlphttp:
    traces_endpoint: "https://pipeline-endpoint.us-east-1.osis.amazonaws.com/
opentelemetry.proto.collector.trace.v1.TraceService/Export"
    auth:
      authenticator: sigv4auth
    compression: none

service:
  extensions: [sigv4auth]
  pipelines:
    traces:
      receivers: [jaeger]
      exporters: [otlphttp]
```

OpenSearchKonfigurasi sampel konsumsi

Untuk mengirim data pelacakan ke domain OpenSearch Service, cobalah konfigurasi pipeline OpenSearch Penyerapan sampel. Untuk petunjuk untuk membuat pipeline, lihat Membuat pipeline Amazon OpenSearch Ingestion.

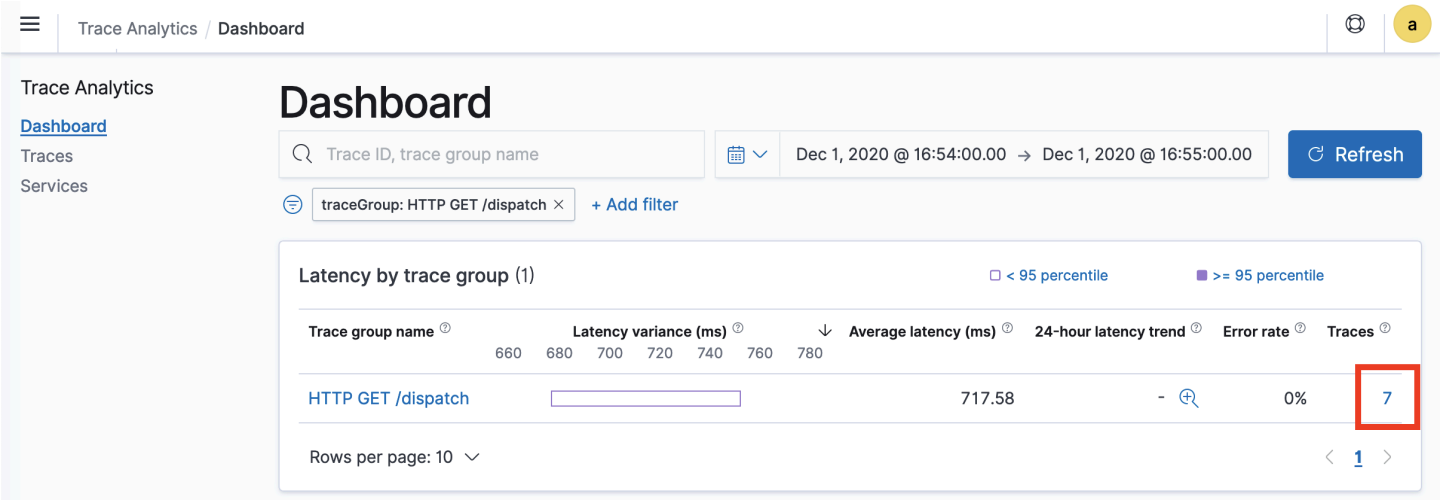
```
version: "2"
otel-trace-pipeline:
  source:
    otel_trace_source:
      "/${pipelineName}/ingest"
  processor:
```

```
    - trace_peer_forwarder:
  sink:
    - pipeline:
        name: "trace_pipeline"
    - pipeline:
        name: "service_map_pipeline"
trace-pipeline:
  source:
    pipeline:
      name: "otel-trace-pipeline"
  processor:
    - otel_traces:
  sink:
    - opensearch:
        hosts: ["https://domain-endpoint"]
        index_type: trace-analytics-raw
        aws:
          # IAM role that OpenSearch Ingestion assumes to access the domain sink
          sts_role_arn: "arn:aws:iam::{account-id}:role/pipeline-role"
          region: "us-east-1"

service-map-pipeline:
  source:
    pipeline:
      name: "otel-trace-pipeline"
  processor:
    - service_map:
  sink:
    - opensearch:
        hosts: ["https://domain-endpoint"]
        index_type: trace-analytics-service-map
        aws:
          # IAM role that the pipeline assumes to access the domain sink
          sts_role_arn: "arn:aws:iam::{account-id}:role/pipeline-role"
          region: "us-east-1"
```

Peran pipeline yang Anda tentukan dalam `sts_role_arn` opsi harus memiliki izin menulis ke wastafel domain. Untuk petunjuk untuk mengonfigurasi izin untuk peran pipeline, lihat Mengizinkan pipeline Amazon OpenSearch Ingestion untuk menulis ke domain.

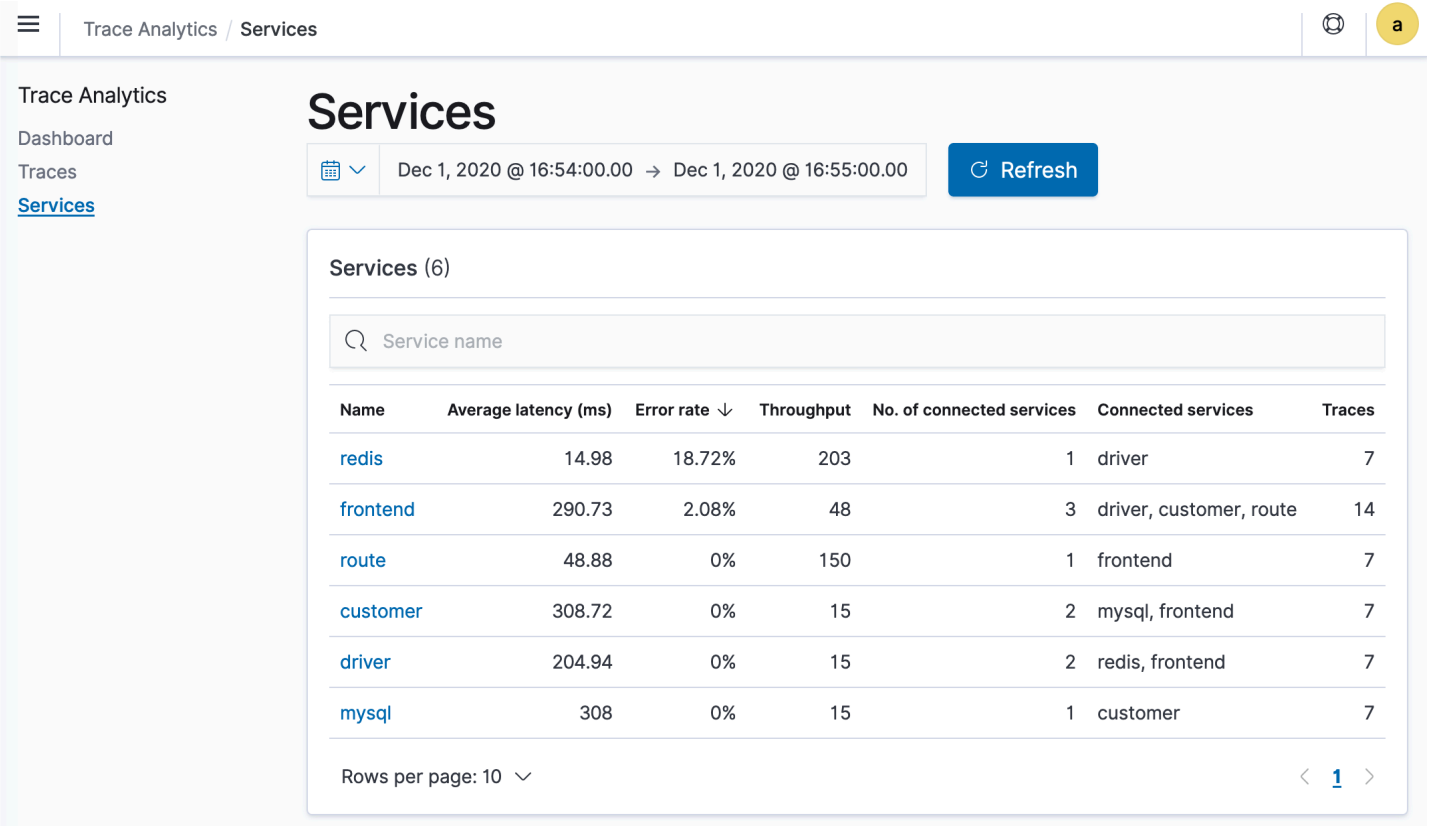
Menjelajahi data pelacakan

Tampilan Dasbor mengelompokkan pelacakan bersama menurut metode dan jalur HTTP sehingga Anda dapat melihat latensi rata-rata, tingkat kesalahan, dan tren yang terkait dengan operasi tertentu. Untuk tampilan yang lebih terfokus, coba pemfilteran berdasarkan nama grup pelacakan.



Untuk menelusuri pelacakan yang membentuk grup pelacakan, pilih jumlah pelacakan di kolom sebelah kanan. Kemudian pilih pelacakan individu untuk ringkasan mendetail.

Tampilan Layanan mencantumkan semua layanan dalam aplikasi, ditambah peta interaktif yang menunjukkan bagaimana berbagai layanan terhubung satu sama lain. Berbeda dengan dasbor (yang membantu mengidentifikasi masalah berdasarkan operasi), peta layanan membantu Anda mengidentifikasi masalah berdasarkan layanan. Cobalah mengurutkan berdasarkan tingkat kesalahan atau latensi untuk merasakan potensi area masalah dari aplikasi Anda.



Membuat kueri data Amazon OpenSearch menggunakan Bahasa Pemrosesan yang Disalurkan

Bahasa Pemrosesan yang Disalurkan (Piped Processing Language-PPL) adalah bahasa kueri yang memungkinkan Anda menggunakan sintaks pipa (|) untuk data kueri yang disimpan di Amazon OpenSearch Service.

Sintaks PPL terdiri dari perintah yang dibatasi oleh karakter pipa (|) di mana data mengalir dari kiri ke kanan melalui setiap alur. Sebagai contoh, sintaks PPL untuk menemukan jumlah host dengan kesalahan HTTP 403 atau 503, agregat mereka per host, dan mengurutkan mereka dalam urutan dampak adalah sebagai berikut:

```
source = dashboards_sample_data_logs | where response='403' or response='503' | stats
 count(request) as request_count by host, response | sort -request_count
```

PPL membutuhkan Elasticsearch 7.9 atau yang lebih baru. OpenSearch Langkah-langkah terperinci dan deskripsi perintah tersedia di manual referensiOpenSearch PPL.

Untuk memulai, pilih Query Workbench di OpenSearch Dasbor dan pilih PPL. Gunakan operasi `bulk` untuk mengindeks beberapa contoh data:

```
PUT accounts/_bulk?refresh
{"index":{"_id":"1"}}
{"account_number":1,"balance":39225,"firstname":"Amber","lastname":"Duke","age":32,"gender":"M"
 Holmes
 Lane","employer":"Pyrami","email":"amberduke@pyrami.com","city":"Brogan","state":"IL"}
{"index":{"_id":"6"}}
{"account_number":6,"balance":5686,"firstname":"Hattie","lastname":"Bond","age":36,"gender":"M"
 Bristol
 Street","employer":"Netagy","email":"hattiebond@netagy.com","city":"Dante","state":"TN"}
{"index":{"_id":"13"}}
{"account_number":13,"balance":32838,"firstname":"Nanette","lastname":"Bates","age":28,"gender"
 Mady Street","employer":"Quility","city":"Nogal","state":"VA"}
{"index":{"_id":"18"}}
{"account_number":18,"balance":4180,"firstname":"Dale","lastname":"Adams","age":33,"gender":"M"
 Hutchinson Court","email":"daleadams@boink.com","city":"Orick","state":"MD"}
```

Contoh berikut mengembalikan bidang `firstname` dan `lastname` untuk dokumen dalam indeks akun dengan `age` lebih besar dari 18:

```
search source=accounts | where age > 18 | fields firstname, lastname
```

Respons Sampel

id	nama depan	nama belakang
0	Amber	Duke
1	Hattie	Bond
2	Nanette	Bates
3	Dale	Adams

Anda dapat menggunakan satu set lengkap perintah baca-saja seperti `search`, `where`, `fields`, `rename`, `dedup`, `stats`, `sort`, `eval`, `head`, `top`, dan `rare`. Untuk deskripsi dan contoh dari setiap perintah, lihat manual referensiOpenSearch PPL.

Plugin PPL mendukung semua fungsi SQL, termasuk matematika, trigonometri, tanggal-waktu, string, agregat, dan operator dan ekspresi canggih. Untuk mempelajari selengkapnya, lihat manual referensiOpenSearch PPL.

Praktik terbaik operasional untuk OpenSearch Layanan Amazon

Bab ini memberikan praktik terbaik untuk mengoperasikan domain OpenSearch Layanan Amazon dan menyertakan pedoman umum yang berlaku untuk banyak kasus penggunaan. Setiap beban kerja unik, dengan karakteristik unik, jadi tidak ada rekomendasi umum yang tepat untuk setiap kasus penggunaan. Praktik terbaik yang paling penting adalah menerapkan, menguji, dan menyetel domain Anda dalam siklus berkelanjutan untuk menemukan konfigurasi, stabilitas, dan biaya yang optimal untuk beban kerja Anda.

Topik

Pemantauan dan peringatan

Praktik terbaik berikut berlaku untuk memantau domain OpenSearch Layanan Anda.

Konfigurasikan CloudWatch alarm

OpenSearch Layanan memancarkan metrik kinerja ke Amazon. CloudWatch Tinjau metrik cluster dan instans Anda secara teratur dan konfigurasikan CloudWatch alarm yang direkomendasikan berdasarkan kinerja beban kerja Anda.

Aktifkan penerbitan log

OpenSearch Layanan mengekspos log OpenSearch kesalahan, mencari log lambat, mengindeks log lambat, dan log audit di Amazon CloudWatch Logs. Cari log lambat, pengindeksan log lambat, dan log kesalahan berguna untuk memecahkan masalah kinerja dan stabilitas. Log audit, yang hanya tersedia jika Anda mengaktifkan kontrol akses berbutir halus untuk melacak aktivitas pengguna. Untuk informasi selengkapnya, lihat Log dalam OpenSearch dokumentasi.

Cari log lambat dan pengindeksan log lambat adalah alat penting untuk memahami dan memecahkan masalah kinerja operasi pencarian dan pengindeksan Anda. Aktifkan pencarian dan indeks pengiriman log lambat untuk semua domain produksi. Anda juga harus mengonfigurasi ambang logging —jika tidak, tidak CloudWatch akan menangkap log.

Strategi pecahan

Pecahan mendistribusikan beban kerja Anda di seluruh node data di domain OpenSearch Layanan Anda. Indeks yang dikonfigurasi dengan benar dapat membantu meningkatkan kinerja domain secara keseluruhan.

Ketika Anda mengirim data ke OpenSearch Layanan, Anda mengirim data tersebut ke indeks. Indeks analog dengan tabel database, dengan dokumen sebagai baris, dan bidang sebagai kolom. Saat Anda membuat indeks, Anda memberi tahu OpenSearch berapa banyak pecahan utama yang ingin Anda buat. Pecahan utama adalah partisi independen dari kumpulan data lengkap. OpenSearch Layanan secara otomatis mendistribusikan data Anda di seluruh pecahan utama dalam indeks. Anda juga dapat mengkonfigurasi replika indeks. Setiap pecahan replika terdiri dari satu set lengkap salinan pecahan utama untuk indeks itu.

OpenSearch Layanan memetakan pecahan untuk setiap indeks di seluruh node data di cluster Anda. Ini memastikan bahwa pecahan primer dan replika untuk indeks berada pada node data yang berbeda. Replika pertama memastikan bahwa Anda memiliki dua salinan data dalam indeks. Anda harus selalu menggunakan setidaknya satu replika. Replika tambahan memberikan redundansi tambahan dan kapasitas baca.

OpenSearch mengirimkan permintaan pengindeksan ke semua node data yang berisi pecahan milik indeks. Ini mengirimkan permintaan pengindeksan pertama ke node data yang berisi pecahan primer, dan kemudian ke node data yang berisi pecahan replika. Permintaan pencarian dirutekan oleh node koordinator ke pecahan primer atau replika untuk semua pecahan milik indeks.

Misalnya, untuk indeks dengan lima pecahan utama dan satu replika, setiap permintaan pengindeksan menyentuh 10 pecahan. Sebaliknya, permintaan pencarian dikirim ke n pecahan, di mana n adalah jumlah pecahan primer. Untuk indeks dengan lima pecahan utama dan satu replika, setiap permintaan pencarian menyentuh lima pecahan (primer atau replika) dari indeks tersebut.

Tentukan jumlah pecahan dan simpul data

Gunakan praktik terbaik berikut untuk menentukan jumlah pecahan dan simpul data untuk domain Anda.

Ukuran pecahan — Ukuran data pada disk adalah hasil langsung dari ukuran data sumber Anda, dan itu berubah saat Anda mengindeks lebih banyak data. source-to-index Rasionya dapat sangat bervariasi, dari 1:10 hingga 10:1 atau lebih, tetapi biasanya sekitar 1:1,10. Anda dapat menggunakan rasio itu untuk memprediksi ukuran indeks pada disk. Anda juga dapat mengindeks beberapa data dan mengambil ukuran indeks aktual untuk menentukan rasio beban kerja Anda. Setelah Anda memiliki ukuran indeks yang diprediksi, tetapkan jumlah pecahan sehingga setiap pecahan akan berada di antara 10—30 GiB (untuk beban kerja pencarian), atau antara 30-50 GiB (untuk beban kerja log). 50 GiB harus maksimal — pastikan untuk merencanakan pertumbuhan.

Jumlah pecahan — Distribusi pecahan ke node data memiliki dampak besar pada kinerja domain. Ketika Anda memiliki indeks dengan beberapa pecahan, cobalah untuk membuat pecahan menghitung kelipatan genap dari jumlah node data. Ini membantu memastikan bahwa pecahan didistribusikan secara merata di seluruh node data, dan mencegah node panas. Misalnya, jika Anda memiliki 12 pecahan primer, jumlah node data Anda harus 2, 3, 4, 6, atau 12. Namun, jumlah pecahan adalah sekunder dari ukuran pecahan — jika Anda memiliki 5 GiB data, Anda tetap harus menggunakan pecahan tunggal.

Pecahan per node data — Jumlah total pecahan yang dapat dipegang oleh node sebanding dengan memori heap Java virtual machine (JVM) node. Bertujuan untuk 25 pecahan atau kurang per GiB memori heap. Misalnya, sebuah node dengan memori heap 32 GiB harus menampung tidak lebih dari 800 pecahan. Meskipun distribusi pecahan dapat bervariasi berdasarkan pola beban kerja Anda, ada batas 1.000 pecahan per node. API kucing/alokasi memberikan tampilan cepat tentang jumlah pecahan dan penyimpanan pecahan total di seluruh node data.

Rasio pecahan terhadap CPU — Ketika pecahan terlibat dalam pengindeksan atau permintaan pencarian, ia menggunakan vCPU untuk memproses permintaan. Sebagai praktik terbaik, gunakan titik skala awal 1,5 vCPU per pecahan. Jika tipe instans Anda memiliki 8 vCPU, atur jumlah node data Anda sehingga setiap node memiliki tidak lebih dari enam pecahan. Perhatikan bahwa ini adalah perkiraan. Pastikan untuk menguji beban kerja Anda dan skala cluster Anda sesuai dengan itu.

Untuk rekomendasi volume penyimpanan, ukuran pecahan, dan jenis instans, lihat sumber daya berikut:

- the section called “Mengukur domain”
- the section called “Menskalakan Petabyte”

Hindari kemiringan penyimpanan

Kemiringan penyimpanan terjadi ketika satu atau lebih node dalam sebuah cluster memegang proporsi penyimpanan yang lebih tinggi untuk satu atau lebih indeks daripada yang lain. Indikasi kemiringan penyimpanan termasuk pemanfaatan CPU yang tidak merata, latensi intermiten dan tidak merata, dan antrian yang tidak merata di seluruh node data. Untuk menentukan apakah Anda memiliki masalah miring, lihat bagian pemecahan masalah berikut:

- the section called “Pecahan simpul dan kemiringan penyimpanan”
- the section called “Pecahan indeks dan kemiringan penyimpanan”

Stabilitas

Praktik terbaik berikut berlaku untuk mempertahankan domain OpenSearch Layanan yang stabil dan sehat.

Tetap terkini dengan OpenSearch

Pembaruan perangkat lunak layanan

OpenSearch Layanan secara teratur merilis pembaruan perangkat lunak yang menambahkan fitur atau meningkatkan domain Anda. Pembaruan tidak mengubah versi mesin OpenSearch atau Elasticsearch. Kami menyarankan Anda menjadwalkan waktu berulang untuk menjalankan operasi DescribeDomainAPI, dan memulai pembaruan perangkat lunak layanan jika ada. `UpdateStatus` `ELIGIBLE` Jika Anda tidak memperbarui domain Anda dalam jangka waktu tertentu (biasanya dua minggu), OpenSearch Layanan akan melakukan pembaruan secara otomatis.

OpenSearch upgrade versi

OpenSearch Layanan secara teratur menambahkan dukungan untuk versi yang dikelola komunitas. OpenSearch Selalu tingkatkan ke OpenSearch versi terbaru saat tersedia.

OpenSearch Layanan secara bersamaan meningkatkan keduanya OpenSearch dan OpenSearch Dasbor (atau Elasticsearch dan Kibana jika domain Anda menjalankan mesin lama). Jika klaster memiliki simpul utama yang didedikasikan, peningkatan selesai tanpa waktu henti. Jika tidak, cluster mungkin tidak responsif selama beberapa detik pasca-peningkatan saat memilih node master. OpenSearch Dasbor mungkin tidak tersedia selama beberapa atau semua peningkatan.

Ada dua cara untuk meng-upgrade domain:

- Upgrade di tempat - Opsi ini lebih mudah karena Anda menyimpan cluster yang sama.
- Snapshot/restore upgrade - Opsi ini bagus untuk menguji versi baru pada cluster baru atau bermigrasi antar cluster.

Terlepas dari proses pemutakhiran yang Anda gunakan, kami menyarankan Anda mempertahankan domain yang semata-mata untuk pengembangan dan pengujian, dan memutakhirkannya ke versi baru sebelum Anda meningkatkan domain produksi Anda. Pilih Pengembangan dan pengujian untuk jenis penerapan saat Anda membuat domain pengujian. Pastikan untuk memutakhirkan semua klien ke versi yang kompatibel segera setelah peningkatan domain.

Tingkatkan kinerja snapshot

Untuk mencegah snapshot Anda macet dalam pemrosesan, jenis instance untuk node master khusus harus sesuai dengan jumlah pecahan. Untuk informasi selengkapnya, lihat the section called “Memilih jenis instance untuk node master khusus”. Selain itu, setiap node harus memiliki tidak lebih dari 25 pecahan yang direkomendasikan per GiB memori heap Java. Untuk informasi selengkapnya, lihat the section called “Memilih jumlah serpihan”.

Aktifkan node master khusus

Node master khusus meningkatkan stabilitas cluster. Master node khusus melakukan tugas manajemen klaster, tetapi tidak menyimpan data indeks atau menanggapi permintaan klien. Pembongkaran tugas manajemen klaster ini meningkatkan stabilitas domain Anda dan memungkinkan beberapa perubahan konfigurasi terjadi tanpa downtime.

Aktifkan dan gunakan tiga node master khusus untuk stabilitas domain optimal di tiga Availability Zone. Menerapkan dengan Multi-AZ dengan Standby mengonfigurasi tiga node master khusus untuk Anda. Misalnya jenis rekomendasi, lihatthe section called “Memilih jenis instance untuk node master khusus”.

Terapkan di beberapa Availability Zone

Untuk mencegah kehilangan data dan meminimalkan downtime cluster jika terjadi gangguan layanan, Anda dapat mendistribusikan node di dua atau tiga Availability Zone secara bersamaan. Wilayah AWSPraktik terbaik adalah menerapkan menggunakan Multi-AZ dengan Siaga, yang mengonfigurasi tiga Availability Zone, dengan dua zona aktif dan satu bertindak sebagai siaga, dan dengan dan dua pecahan replika per indeks. Konfigurasi ini memungkinkan OpenSearch Service mendistribusikan pecahan replika ke AZ yang berbeda dari pecahan primer yang sesuai. Tidak ada biaya transfer data lintas-AZ untuk komunikasi klaster antara Availability Zones.

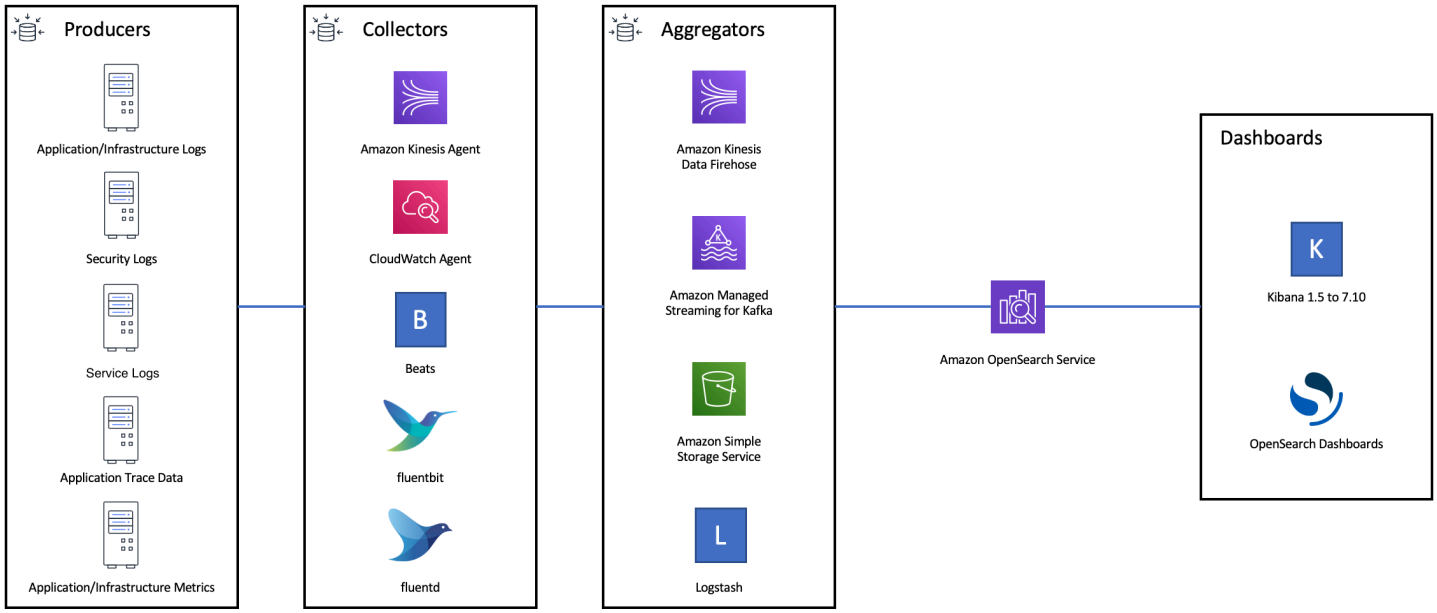
Availability Zone adalah lokasi terisolasi di setiap Wilayah. Dengan konfigurasi dua-AZ, kehilangan satu Availability Zone berarti Anda kehilangan setengah dari semua kapasitas domain. Pindah ke tiga Availability Zone semakin mengurangi dampak kehilangan satu Availability Zone.

Kontrol aliran menelan dan buffering

Kami menyarankan Anda membatasi jumlah permintaan keseluruhan menggunakan operasi _bulk API. Lebih efisien untuk mengirim satu `_bulk` permintaan yang berisi 5.000 dokumen daripada mengirim 5.000 permintaan yang berisi satu dokumen.

Untuk stabilitas operasional yang optimal, terkadang perlu membatasi atau bahkan menjeda aliran hulu permintaan pengindeksan. Membatasi tingkat permintaan indeks adalah mekanisme penting untuk menangani lonjakan permintaan yang tidak terduga atau sesekali yang mungkin membanjiri klaster. Pertimbangkan untuk membangun mekanisme kontrol aliran ke dalam arsitektur hulu Anda.

Diagram berikut menunjukkan beberapa opsi komponen untuk arsitektur log ingest. Konfigurasikan lapisan agregasi untuk memungkinkan ruang yang cukup untuk menyangga data yang masuk untuk lonjakan lalu lintas mendadak dan pemeliharaan domain singkat.



Buat pemetaan untuk beban kerja penelusuran

Untuk beban kerja pencarian, buat pemetaan yang menentukan cara OpenSearch menyimpan dan mengindeks dokumen dan bidangnya. Setel `dynamic` ke `strict` untuk mencegah bidang baru ditambahkan secara tidak sengaja.

```
PUT my-index
{
  "mappings": {
    "dynamic": "strict",
    "properties": {
      "title": { "type" : "text" },
      "author": { "type" : "integer" },
      "year": { "type" : "text" }
    }
  }
}
```

Gunakan templat indeks

Anda dapat menggunakan templat indeks sebagai cara untuk memberi tahu OpenSearch cara mengonfigurasi indeks saat dibuat. Konfigurasikan template indeks sebelum membuat indeks. Kemudian, ketika Anda membuat indeks, itu mewarisi pengaturan dan pemetaan dari template. Anda dapat menerapkan lebih dari satu templat ke satu indeks, sehingga Anda dapat menentukan pengaturan di satu templat dan pemetaan di templat lainnya. Strategi ini memungkinkan satu templat

untuk pengaturan umum di beberapa indeks, dan templat terpisah untuk pengaturan dan pemetaan yang lebih spesifik.

Pengaturan berikut sangat membantu untuk mengkonfigurasi dalam template:

- Jumlah pecahan primer dan replika
- Interval penyegaran (seberapa sering menyegarkan dan membuat perubahan terbaru pada indeks yang tersedia untuk dicari)
- Kontrol pemetaan dinamis
- Pemetaan bidang eksplisit

Contoh template berikut berisi masing-masing pengaturan ini:

```
{
   "index_patterns":[
      "index-*"
   ],
   "order": 0,
   "settings": {
      "index": {
         "number_of_shards": 3,
         "number_of_replicas": 1,
         "refresh_interval": "60s"
      }
   },
   "mappings": {
      "dynamic": false,
      "properties": {
         "field_name1": {
            "type": "keyword"
         }
      }
   }
}
```

Bahkan jika mereka jarang berubah, memiliki pengaturan dan pemetaan yang ditentukan secara OpenSearch terpusat lebih sederhana untuk dikelola daripada memperbarui beberapa klien hulu.

Mengelola indeks dengan Index State Management

Jika Anda mengelola log atau data deret waktu, sebaiknya gunakan Index State Management (ISM). ISM memungkinkan Anda mengotomatiskan tugas manajemen siklus hidup indeks reguler. Dengan ISM, Anda dapat membuat kebijakan yang memanggil rollover alias indeks, mengambil snapshot indeks, memindahkan indeks antar tingkatan penyimpanan, dan menghapus indeks lama. Anda bahkan dapat menggunakan operasi rollover ISM sebagai strategi manajemen siklus hidup data alternatif untuk menghindari shard skew.

Pertama, buat kebijakan ISM. Sebagai contoh, lihat the section called “Contoh kebijakan”. Kemudian, lampirkan kebijakan ke satu atau lebih indeks. Jika Anda menyertakan bidang templat ISM dalam kebijakan, OpenSearch Layanan secara otomatis menerapkan kebijakan tersebut ke indeks apa pun yang cocok dengan pola yang ditentukan.

Hapus indeks yang tidak digunakan

Tinjau indeks di klaster Anda secara teratur dan identifikasi indeks apa pun yang tidak digunakan. Ambil snapshot dari indeks tersebut sehingga disimpan di S3, lalu hapus. Saat Anda menghapus indeks yang tidak digunakan, Anda mengurangi jumlah pecahan, dan memungkinkan distribusi penyimpanan dan pemanfaatan sumber daya yang lebih seimbang di seluruh node. Bahkan ketika mereka menganggur, indeks mengkonsumsi beberapa sumber daya selama kegiatan pemeliharaan indeks internal.

Daripada menghapus indeks yang tidak digunakan secara manual, Anda dapat menggunakan ISM untuk secara otomatis mengambil snapshot dan menghapus indeks setelah jangka waktu tertentu.

Gunakan beberapa domain untuk ketersediaan tinggi

Untuk mencapai ketersediaan tinggi di luar waktu aktif 99,9% di beberapa Wilayah, pertimbangkan untuk menggunakan dua domain. Untuk kumpulan data kecil atau lambat berubah, Anda dapat mengatur replikasi lintas cluster untuk mempertahankan model aktif-pasif. Dalam model ini, hanya domain pemimpin yang ditulis, tetapi salah satu domain dapat dibaca. Untuk kumpulan data yang lebih besar dan data yang berubah dengan cepat, konfigurasikan pengiriman ganda di pipeline ingest Anda sehingga semua data ditulis secara independen ke kedua domain dalam model aktif-aktif.

Arsitek aplikasi hulu dan hilir Anda dengan mempertimbangkan failover. Pastikan untuk menguji proses failover bersama dengan proses pemulihan bencana lainnya.

Performa

Praktik terbaik berikut berlaku untuk menyetel domain Anda untuk kinerja yang optimal.

Optimalkan ukuran dan kompresi permintaan massal

Ukuran massal tergantung pada data, analisis, dan konfigurasi cluster Anda, tetapi titik awal yang baik adalah 3-5 MiB per permintaan massal.

Kirim permintaan dan terima tanggapan dari OpenSearch domain Anda dengan menggunakan kompresi gzip untuk mengurangi ukuran payload permintaan dan tanggapan. Anda dapat menggunakan kompresi gzip dengan klien OpenSearch Python, atau dengan menyertakan header berikut dari sisi klien:

- `'Accept-Encoding': 'gzip'`
- `'Content-Encoding': 'gzip'`

Untuk mengoptimalkan ukuran permintaan massal Anda, mulailah dengan ukuran permintaan massal 3 MiB. Kemudian, perlahan-lahan tingkatkan ukuran permintaan hingga kinerja pengindeksan berhenti membaik.

> **Note**
>
> Untuk mengaktifkan kompresi gzip pada domain yang menjalankan Elasticsearch versi 6.x, Anda harus mengatur `http_compression.enabled` pada tingkat cluster. Pengaturan ini benar secara default di Elasticsearch versi 7.x dan semua versi. OpenSearch

Mengurangi ukuran respons permintaan massal

Untuk mengurangi ukuran OpenSearch respons, kecualikan bidang yang tidak perlu dengan `filter_path` parameter. Pastikan Anda tidak memfilter bidang apa pun yang diperlukan untuk mengidentifikasi atau mencoba ulang permintaan yang gagal. Untuk informasi selengkapnya dan contoh tambahan, lihat the section called “Mengurangi ukuran respons”.

Selaraskan interval penyegaran

OpenSearch indeks akhirnya memiliki konsistensi baca. Operasi penyegaran membuat semua pembaruan yang dilakukan pada indeks tersedia untuk pencarian. Interval penyegaran default adalah satu detik, yang berarti OpenSearch melakukan penyegaran setiap detik saat indeks sedang ditulis.

Semakin jarang Anda menyegarkan indeks (interval penyegaran yang lebih tinggi), semakin baik kinerja pengindeksan keseluruhan. Trade-off dari peningkatan interval penyegaran adalah bahwa ada penundaan yang lebih lama antara pembaruan indeks dan ketika data baru tersedia untuk pencarian. Atur interval penyegaran setinggi yang dapat Anda toleransi untuk meningkatkan kinerja secara keseluruhan.

Sebaiknya atur `refresh_interval` parameter untuk semua indeks Anda menjadi 30 detik atau lebih.

Aktifkan Auto-Tune

Auto-Tune menggunakan metrik kinerja dan penggunaan dari OpenSearch klaster Anda untuk menyarankan perubahan ukuran antrian, ukuran cache, dan pengaturan mesin virtual Java (JVM) pada node Anda. Perubahan opsional ini meningkatkan kecepatan dan stabilitas klaster. Anda dapat kembali ke pengaturan OpenSearch Layanan default kapan saja. Auto-Tune diaktifkan secara default pada domain baru kecuali Anda menonaktifkannya secara eksplisit.

Kami menyarankan Anda mengaktifkan Auto-Tune di semua domain, dan mengatur jendela pemeliharaan berulang atau meninjau rekomendasinya secara berkala.

Keamanan

Praktik terbaik berikut berlaku untuk mengamankan domain Anda.

Aktifkan kontrol akses berbutir halus

Kontrol akses berbutir halus memungkinkan Anda mengontrol siapa yang dapat mengakses data tertentu dalam domain Layanan. OpenSearch Dibandingkan dengan kontrol akses umum, kontrol akses berbutir halus memberikan setiap klaster, indeks, dokumen, dan bidang kebijakan yang ditentukan sendiri untuk akses. Kriteria akses dapat didasarkan pada sejumlah faktor, termasuk peran orang yang meminta akses dan tindakan yang ingin mereka lakukan pada data. Misalnya, Anda mungkin memberi satu pengguna akses untuk menulis ke indeks, dan akses pengguna lain hanya untuk membaca data pada indeks tanpa membuat perubahan apa pun.

Kontrol akses berbutir halus memungkinkan data dengan persyaratan akses yang berbeda ada di ruang penyimpanan yang sama tanpa mengalami masalah keamanan atau kepatuhan.

Sebaiknya aktifkan kontrol akses berbutir halus pada domain Anda.

Menyebarkan domain dalam VPC

Menempatkan domain OpenSearch Layanan Anda dalam virtual private cloud (VPC) membantu memungkinkan komunikasi yang aman antara OpenSearch Layanan dan layanan lain dalam VPC — tanpa memerlukan gateway internet, perangkat NAT, atau koneksi VPN. Semua lalu lintas tetap aman di dalam AWS Cloud. Karena isolasi logisnya, domain yang berada di dalam VPC memiliki lapisan keamanan ekstra dibandingkan dengan domain yang menggunakan titik akhir publik.

Kami menyarankan Anda membuat domain Anda dalam VPC.

Menerapkan kebijakan akses terbatas

Bahkan jika domain Anda digunakan dalam VPC, itu adalah praktik terbaik untuk menerapkan keamanan secara berlapis-lapis. Pastikan untuk memeriksa konfigurasi kebijakan akses Anda saat ini.

Terapkan kebijakan akses berbasis sumber daya terbatas ke domain Anda dan ikuti prinsip hak istimewa paling sedikit saat memberikan akses ke API konfigurasi dan operasi API. OpenSearch Sebagai aturan umum, hindari penggunaan prinsipal pengguna anonim `"Principal": {"AWS": "*" }` dalam kebijakan akses Anda.

Namun, ada beberapa situasi di mana penggunaan kebijakan akses terbuka dapat diterima, seperti saat Anda mengaktifkan kontrol akses berbutir halus. Kebijakan akses terbuka dapat memungkinkan Anda mengakses domain jika penandatanganan permintaan sulit atau tidak mungkin dilakukan, seperti dari klien dan alat tertentu.

Aktifkan enkripsi saat istirahat

OpenSearch Domain layanan menawarkan enkripsi data saat istirahat untuk membantu mencegah akses tidak sah ke data Anda. Enkripsi saat istirahat menggunakan AWS Key Management Service (AWS KMS) untuk menyimpan dan mengelola kunci enkripsi Anda, dan algoritma Advanced Encryption Standard dengan kunci 256-bit (AES-256) untuk melakukan enkripsi.

Jika domain Anda menyimpan data sensitif, aktifkan enkripsi data saat istirahat.

Aktifkan node-to-node enkripsi

ode-to-node Enkripsi N menyediakan lapisan keamanan tambahan di atas fitur keamanan default dalam OpenSearch Layanan. Ini mengimplementasikan Transport Layer Security (TLS) untuk semua komunikasi antara node yang disediakan di dalamnya. OpenSearch ode-to-node Enkripsi N, data apa pun yang dikirim ke domain OpenSearch Layanan Anda melalui HTTPS tetap dienkripsi dalam perjalanan saat sedang didistribusikan dan direplikasi antar node.

Jika domain Anda menyimpan data sensitif, aktifkan node-to-node enkripsi.

Monitor dengan AWS Security Hub

Pantau penggunaan OpenSearch Layanan Anda yang berkaitan dengan praktik terbaik keamanan dengan menggunakan AWS Security Hub. Hub Keamanan menggunakan kontrol keamanan untuk mengevaluasi konfigurasi sumber daya dan standar keamanan guna membantu Anda mematuhi berbagai kerangka kerja kepatuhan. Untuk informasi selengkapnya tentang penggunaan Security Hub guna mengevaluasi sumber daya OpenSearch Layanan, lihat Amazon OpenSearch Service kontrol di PanduanAWS Security Hub Pengguna.

Optimasi biaya

Praktik terbaik berikut berlaku untuk mengoptimalkan dan menghemat biaya OpenSearch Layanan Anda.

Gunakan jenis instans generasi terbaru

OpenSearch Layanan selalu mengadopsi jenis instans Amazon EC2 baru yang memberikan kinerja lebih baik dengan biaya lebih rendah. Kami merekomendasikan untuk selalu menggunakan instance generasi terbaru.

Hindari penggunaan T2 atau `t3.small` instance untuk domain produksi karena mereka dapat menjadi tidak stabil di bawah beban berat yang berkelanjutan. `t3.medium`instance adalah opsi untuk beban kerja produksi kecil (baik sebagai node data dan sebagai node master khusus).

Gunakan volume gp3 Amazon EBS terbaru

OpenSearch node data membutuhkan latensi rendah dan penyimpanan throughput tinggi untuk menyediakan pengindeksan dan kueri yang cepat. Dengan menggunakan volume Amazon EBS gp3,

Anda mendapatkan kinerja baseline yang lebih tinggi (IOPS dan throughput) dengan biaya 9,6% lebih rendah dibandingkan dengan jenis volume Amazon EBS gp2 yang ditawarkan sebelumnya. Anda dapat menyediakan IOPS tambahan dan throughput independen dari ukuran volume menggunakan gp3. Volume ini juga lebih stabil daripada volume generasi sebelumnya karena tidak menggunakan kredit burst. Jenis volume gp3 juga menggandakan batas ukuran per-data-node volume tipe volume gp2. Dengan volume yang lebih besar ini, Anda dapat mengurangi biaya data pasif dengan meningkatkan jumlah penyimpanan per node data.

Penggunaan UltraWarm dan penyimpanan dingin untuk data log deret waktu

Jika Anda menggunakan OpenSearch analisis log, pindahkan data Anda ke UltraWarm atau penyimpanan dingin untuk mengurangi biaya. Gunakan Index State Management (ISM) untuk memigrasikan data antar tingkatan penyimpanan dan mengelola retensi data.

UltraWarmmenyediakan cara yang hemat biaya untuk menyimpan sejumlah besar data hanya-baca di Layanan. OpenSearch UltraWarm menggunakan Amazon S3 untuk penyimpanan, yang berarti bahwa data tidak dapat diubah dan hanya satu salinan yang diperlukan. Anda hanya membayar untuk penyimpanan yang setara dengan ukuran pecahan utama dalam indeks Anda. Latensi untuk UltraWarm kueri tumbuh dengan jumlah data S3 yang diperlukan untuk melayani kueri. Setelah data di-cache pada node, kueri ke UltraWarm indeks berkinerja mirip dengan kueri ke indeks panas.

Cold storage juga didukung oleh S3. Saat Anda perlu menanyakan data dingin, Anda dapat secara selektif melampirkannya ke UltraWarm node yang ada. Data dingin menimbulkan biaya penyimpanan terkelola yang sama seperti UltraWarm, tetapi objek dalam penyimpanan dingin tidak mengkonsumsi sumber daya UltraWarm node. Oleh karena itu, cold storage menyediakan sejumlah besar kapasitas penyimpanan tanpa mempengaruhi ukuran atau jumlah UltraWarm node.

UltraWarm menjadi hemat biaya ketika Anda memiliki sekitar 2,5 TiB data untuk bermigrasi dari penyimpanan panas. Pantau laju pengisian Anda dan rencanakan untuk memindahkan indeks UltraWarm sebelum Anda mencapai volume data tersebut.

Meninjau rekomendasi untuk Instans Cadangan

Pertimbangkan untuk membeli Instans Cadangan (RI) setelah Anda memiliki dasar yang baik pada kinerja dan konsumsi komputasi Anda. Diskon mulai dari sekitar 30% untuk reservasi 1 tahun tanpa di muka dan dapat meningkat hingga 50% untuk komitmen 3 tahun di muka.

Setelah Anda mengamati operasi stabil setidaknya selama 14 hari, tinjau rekomendasi Instans Cadangan di Cost Explorer. Judul OpenSearch Layanan Amazon menampilkan rekomendasi pembelian RI spesifik dan penghematan yang diproyeksikan.

Mengukur domain OpenSearch Layanan Amazon

Tidak ada metode sempurna untuk mengukur domain OpenSearch Layanan Amazon. Namun, dengan memulai dengan pemahaman tentang kebutuhan penyimpanan Anda, layanan, dan OpenSearch dirinya sendiri, Anda dapat membuat perkiraan awal yang terdidik tentang kebutuhan perangkat keras Anda. Perkiraan ini dapat berfungsi sebagai titik awal yang berguna untuk aspek paling penting dari ukuran domain: mengujinya dengan beban kerja yang representatif dan memantau performanya.

Topik

Menghitung persyaratan penyimpanan

Sebagian besar OpenSearch beban kerja termasuk dalam salah satu dari dua kategori besar:

- Indeks berumur panjang: Anda menulis kode yang memproses data menjadi satu atau lebih OpenSearch indeks dan kemudian memperbarui indeks tersebut secara berkala saat data sumber berubah. Beberapa contoh umum adalah situs web, dokumen, dan pencarian e-commerce.
- Indeks bergulir: Data terus mengalir ke satu set indeks sementara, dengan periode pengindeksan dan jendela retensi (seperti sekumpulan indeks harian yang dipertahankan selama dua minggu). Beberapa contoh umum adalah analitik log, pemrosesan seri waktu, dan analitik aliran klik.

Untuk beban kerja indeks berumur panjang, Anda dapat memeriksa sumber data pada disk dan dengan mudah menentukan berapa banyak ruang penyimpanan mengonsumsinya. Jika data berasal dari berbagai sumber, cukup tambahkan sumber tersebut bersama-sama.

Untuk indeks bergulir, Anda dapat mengalikan jumlah data yang dihasilkan selama periode waktu yang representatif dengan periode retensi. Misalnya, jika Anda menghasilkan 200 MiB data log per

jam, itu adalah 4,7 GiB per hari, yaitu 66 GiB data pada waktu tertentu jika Anda memiliki periode retensi dua minggu.

Namun, ukuran data sumber Anda hanyalah salah satu aspek dari kebutuhan penyimpanan Anda. Anda juga harus mempertimbangkan hal berikut:

- Jumlah replika: Setiap replika adalah salinan lengkap indeks dan kebutuhan jumlah ruang disk yang sama. Secara default, setiap OpenSearch indeks memiliki satu replika. Kami merekomendasikan setidaknya satu replika untuk mencegah kehilangan data. Replika juga meningkatkan performa pencarian, sehingga Anda mungkin ingin replika lebih banyak jika Anda memiliki beban kerja baca-berat. Gunakan `PUT /my-index/_settings` untuk memperbarui pengaturan `number_of_replicas` untuk indeks Anda.
- OpenSearch overhead pengindeksan: Ukuran indeks pada disk bervariasi. Ukuran total data sumber ditambah indeks seringkali 110% dari sumber, dengan indeks hingga 10% dari data sumber. Setelah mengindeks data, Anda dapat menggunakan `_cat/indices?v` API dan `pri.store.size` nilai untuk menghitung overhead yang tepat. `_cat/allocation?v`juga memberikan ringkasan yang berguna.
- Ruang cadangan sistem operasi yang disediakan: Secara default, Linux mencadangkan 5% dari sistem file untuk `root` pengguna guna proses kritis, pemulihan sistem, dan untuk melindungi terhadap masalah fragmentasi disk.
- OpenSearch Layanan overhead: OpenSearch Layanan mencadangkan 20% dari ruang penyimpanan setiap instans (hingga 20 GiB) untuk penggabungan segmen, log, dan operasi internal lainnya.

  Karena maksimum 20 GiB ini, jumlah total ruang yang dicadangkan dapat bervariasi secara dramatis tergantung pada jumlah instans di domain Anda. Sebagai contoh, sebuah domain mungkin memiliki tiga instans `m6g.xlarge.search`, masing-masing dengan 500 GIB ruang penyimpanan, dengan total 1,46 TiB. Dalam hal ini, total ruang yang dicadangkan hanya 60 GiB. Domain lainnya mungkin memiliki 10 instans `m3.medium.search`, masing-masing dengan 100 GIB ruang penyimpanan, dengan total 0,98 TiB. Di sini, total ruang yang dicadangkan adalah 200 GiB, meskipun domain pertama adalah 50% lebih besar.

  Dalam rumus berikut, kami menerapkan perkiraan “kasus terburuk” untuk overhead. Perkiraan ini mencakup ruang kosong tambahan untuk membantu meminimalkan dampak kegagalan node dan pemadaman Availability Zone.

Singkatnya, jika Anda memiliki 66 GiB data pada waktu tertentu dan ingin satu replika, persyaratan penyimpanan minimum lebih dekat dengan 66 * 2 * 1.1 / 0.95 / 0.8 = 191 GiB. Anda dapat menggeneralisasi perhitungan ini sebagai berikut:

Data sumber* (1 + jumlah replika) * (1 + overhead pengindeksan)/(1 - Ruang cadangan Linux)/(1 - Overhead OpenSearch layanan) = persyaratan penyimpanan minimum

Atau Anda dapat menggunakan versi yang disederhanakan ini:

Sumber data* (1 + jumlah replika) * 1,45 = persyaratan penyimpanan minimum

Ruang penyimpanan yang tidak mencukupi adalah salah satu penyebab paling umum dari ketidakstabilan cluster. Jadi, Anda harus memeriksa ulang angka ketika Anda memilih jenis instance, jumlah instance, dan volume penyimpanan.

Pertimbangan penyimpanan lainnya ada:

- Jika persyaratan penyimpanan minimum melebihi 1 PB, lihat the section called “Menskalakan Petabyte”.
- Jika Anda memiliki indeks bergulir dan ingin menggunakan arsitektur panas-hangat, lihat. the section called “UltraWarm penyimpanan”

Memilih jumlah serpihan

Setelah memahami persyaratan penyimpanan, Anda dapat menyelidiki strategi pengindeksan. Secara default di OpenSearch Layanan, setiap indeks dibagi menjadi lima pecahan utama dan satu replika (total 10 pecahan). Perilaku ini berbeda dari open source OpenSearch, yang default ke satu pecahan primer dan satu replika. Karena Anda tidak dapat dengan mudah mengubah jumlah serpihan primer untuk indeks yang ada, Anda harus memutuskan tentang jumlah serpihan sebelum mengindeks dokumen pertama Anda.

Tujuan keseluruhan memilih sejumlah pecahan adalah untuk mendistribusikan indeks secara merata di semua node data di cluster. Namun, serpihan ini seharusnya tidak terlalu besar atau terlalu banyak. Pedoman umum adalah mencoba menjaga ukuran pecahan antara 10-30 GiB untuk beban kerja di mana latensi pencarian adalah tujuan kinerja utama, dan 30-50 GiB untuk beban kerja berat tulis seperti analitik log.

Pecahan besar dapat menyulitkan OpenSearch untuk pulih dari kegagalan, tetapi karena setiap pecahan menggunakan sejumlah CPU dan memori, memiliki terlalu banyak pecahan kecil dapat

menyebabkan masalah kinerja dan kesalahan memori. Dengan kata lain, pecahan harus cukup kecil sehingga instance OpenSearch Service yang mendasarinya dapat menanganinya, tetapi tidak terlalu kecil sehingga mereka menempatkan tekanan yang tidak perlu pada perangkat keras.

Misalnya, anggap Anda memiliki 66 GiB data. Anda tidak mengharapkan angka itu meningkat dari waktu ke waktu, dan Anda ingin menyimpan serpihan Anda sekitar 30 GiB masing-masing. Oleh karena itu, jumlah serpihan Anda harus kira-kira 66 * 1,1/30 = 3. Anda dapat menggeneralisasi perhitungan ini sebagai berikut:

(Data sumber+ruang untuk tumbuh) * (1 + overhead pengindeksan)/ukuran pecahan yang diinginkan = perkiraan jumlah pecahan primer

Persamaan ini membantu mengimbangi pertumbuhan data dari waktu ke waktu. Jika Anda mengharapkan 66 GiB data yang sama menjadi empat kali lipat selama tahun depan, perkiraan jumlah serpihan adalah (66 + 198) * 1,1/30 = 10. Ingat, Anda belum memiliki data ekstra 198 GiB data. Periksa untuk memastikan bahwa persiapan untuk masa depan ini tidak membuat serpihan kecil yang tidak perlu yang menghabiskan banyak CPU dan memori saat ini. Dalam hal ini, 66 * 1,1 / 10 serpihan = 7,26 GiB per serpihan, yang akan mengonsumsi sumber daya tambahan dan berada di bawah kisaran ukuran yang disarankan. Anda dapat mempertimbangkan middle-of-the-road pendekatan lebih dari enam pecahan, yang membuat Anda memiliki pecahan 12-GiB hari ini dan pecahan 48-GiB di masa depan. Kemudian lagi, Anda mungkin lebih memilih untuk memulai dengan tiga serpihan dan mengindeks ulang data Anda ketika serpihan melebihi 50 GiB.

Masalah yang jauh lebih jarang melibatkan pembatasan jumlah serpihan per simpul. Jika Anda ukuran serpihan Anda tepat, Anda biasanya kehabisan ruang disk lama sebelum menghadapi batas ini. Misalnya, instans `m6g.large.search` memiliki ukuran disk maksimum 512 GiB. Jika Anda tetap di bawah 80% penggunaan disk dan ukuran serpihan Anda pada 20 GiB, maka dapat menampung sekitar 20 serpihan. Elasticsearch 7. x dan yang lebih baru, dan semua versi OpenSearch, memiliki batas 1.000 pecahan per node. Untuk menyesuaikan pecahan maksimum per node, konfigurasikan `cluster.max_shards_per_node` pengaturan. Sebagai contoh, lihat Pengaturan cluster.

Mengukur serpihan dengan tepat hampir selalu membuat Anda berada di bawah batas ini, tetapi Anda juga dapat mempertimbangkan jumlah serpihan untuk setiap GiB heap Java. Pada node tertentu, memiliki tidak lebih dari 25 pecahan per GiB heap Java. Misalnya, sebuah `m5.large.search` instance memiliki tumpukan 4-GiB, sehingga setiap node harus memiliki tidak lebih dari 100 pecahan. Pada hitungan serpihan tersebut, setiap serpihan berukuran sekitar 5 GiB, yang jauh di bawah rekomendasi kami.

Memilih tipe instans dan pengujian

Setelah menghitung kebutuhan penyimpanan dan memilih jumlah serpihan yang Anda butuhkan, Anda dapat mulai membuat keputusan perangkat keras. Persyaratan perangkat keras bervariasi secara dramatis oleh beban kerja, tetapi kami masih dapat menawarkan beberapa rekomendasi dasar.

Secara umum, batas penyimpanan untuk setiap tipe instans dipetakan dengan jumlah CPU dan memori yang mungkin Anda butuhkan untuk beban kerja ringan. Misalnya, instans `m6g.large.search` memiliki ukuran volume EBS maksimum 512 GiB, 2 core vCPU, dan 8 GiB memori. Jika klaster Anda memiliki banyak serpihan, melakukan pajak agregasi, sering memperbarui dokumen, atau memproses sejumlah besar kueri, sumber daya tersebut mungkin tidak cukup untuk kebutuhan Anda. Jika cluster Anda termasuk dalam salah satu kategori ini, coba mulai dengan konfigurasi yang mendekati 2 core vCPU dan 8 GiB memori untuk setiap 100 GiB dari kebutuhan penyimpanan Anda.

> **Tip**
>
> Untuk ringkasan sumber daya perangkat keras yang dialokasikan ke setiap jenis instans, lihat harga OpenSearch Layanan Amazon.

Namun, bahkan sumber daya tersebut mungkin tidak cukup. Beberapa OpenSearch pengguna melaporkan bahwa mereka membutuhkan sumber daya berkali-kali untuk memenuhi persyaratan mereka. Untuk menemukan perangkat keras yang tepat untuk beban kerja Anda, Anda harus membuat perkiraan awal yang terdidik, menguji dengan beban kerja yang representatif, menyesuaikan, dan menguji lagi.

Langkah 1: Buat anggaran awal

Untuk memulai, kami merekomendasikan minimal tiga node untuk menghindari OpenSearch masalah potensial, seperti keadaan otak terbelah (ketika selang komunikasi mengarah ke cluster yang memiliki dua node master). Jika Anda memiliki tiga simpul utama khusus, kami masih merekomendasikan minimal dua data simpul untuk replikasi.

Langkah 2: Hitung persyaratan penyimpanan per simpul

Jika Anda memiliki persyaratan penyimpanan 184-GiB dan jumlah minimum tiga node yang disarankan, gunakan persamaan 184/3 = 61 GiB untuk menemukan jumlah penyimpanan yang

dibutuhkan setiap node. Dalam contoh ini, Anda dapat memilih tiga `m6g.large.search` contoh, di mana masing-masing menggunakan volume penyimpanan EBS 90-GiB, sehingga Anda memiliki jaring pengaman dan beberapa ruang untuk pertumbuhan dari waktu ke waktu. Konfigurasi ini menyediakan 6 core vCPU dan memori 24 GiB, sehingga cocok untuk beban kerja yang lebih ringan.

Untuk contoh yang lebih penting, pertimbangkan persyaratan penyimpanan 14 TiB (14.336 GiB) dan beban kerja yang berat. Dalam hal ini, Anda dapat memilih untuk memulai pengujian dengan 2 * 144 = 288 core vCPU dan 8 * 144 = 1152 GiB memori. Angka-angka ini bekerja untuk sekitar 18 instans `i3.4xlarge.search`. Jika Anda tidak memerlukan penyimpanan lokal yang cepat, Anda juga dapat menguji 18 `r6g.4xlarge.search` instans, masing-masing menggunakan volume penyimpanan EBS 1-TiB.

Jika klaster Anda mencakup ratusan terabyte data, lihat the section called “Menskalakan Petabyte”.

Langkah 3: Lakukan pengujian perwakilan

Setelah mengonfigurasi klaster, Anda dapat menambahkan indeks menggunakan jumlah pecahan yang Anda hitung sebelumnya, melakukan beberapa pengujian klien representatif menggunakan kumpulan data realistis, dan memantau CloudWatch metrik untuk melihat bagaimana klaster menangani beban kerja.

Langkah 4: Berhasil atau ulangi

Jika kinerja memenuhi kebutuhan Anda, pengujian berhasil, dan CloudWatch metrik normal, cluster siap digunakan. Ingatlah untuk mengatur CloudWatch alarm untuk mendeteksi penggunaan sumber daya yang tidak sehat.

Jika performa tidak dapat diterima, percobaan gagal, atau `CPUUtilization` atau `JVMMemoryPressure` tinggi, Anda mungkin perlu memilih tipe instans yang berbeda (atau menambahkan instans) dan melanjutkan pengujian. Saat Anda menambahkan instance, OpenSearch secara otomatis menyeimbangkan kembali distribusi pecahan di seluruh cluster.

Karena lebih mudah untuk mengukur kelebihan kapasitas dalam cluster yang dikuasai daripada defisit pada cluster yang kurang bertenaga, kami sarankan memulai dengan cluster yang lebih besar dari yang Anda pikir Anda butuhkan. Selanjutnya, uji dan turunkan skala ke klaster efisien yang memiliki sumber daya ekstra untuk memastikan operasi yang stabil selama periode peningkatan aktivitas.

Klaster produksi atau klaster dengan status kompleks mendapat manfaat dari simpul utama khusus, yang meningkatkan performa dan keandalan klaster.

Skala petabyte di Layanan Amazon OpenSearch

Domain Amazon OpenSearch Service menawarkan penyimpanan terlampir hingga 3 PB. Anda dapat mengonfigurasi domain dengan 200 tipe instans `i3.16xlarge.search`, masing-masing dengan penyimpanan 15 TB. Karena perbedaan tipis dalam skala, rekomendasi untuk domain ukuran ini berbeda dari rekomendasi umum kami. Bagian ini membahas pertimbangan untuk menciptakan domain, biaya, penyimpanan, dan ukuran serpihan.

Sementara bagian ini sering mereferensikan tipe instans `i3.16xlarge.search`, Anda dapat menggunakan beberapa tipe instans lain untuk mencapai 1 PB dari total penyimpanan domain.

Membuat domain

Domain dengan ukuran ini melebihi batas default 80 instance per domain. Untuk meminta peningkatan batas layanan hingga 200 instans per domain, buka kasus di AWS Pusat Dukungan.

Harga

Sebelum membuat domain sebesar ini, periksa halaman harga OpenSearch Layanan Amazon untuk memastikan bahwa biaya terkait sesuai dengan harapan Anda. Memeriksa the section called “UltraWarm penyimpanan” untuk melihat apakah arsitektur hangat hangat cocok dengan kasus penggunaan Anda.

Penyimpanan

Tipe instans `i3` dirancang untuk menyediakan penyimpanan cepat, penyimpanan non-volatile memory express (NVMe). Karena penyimpanan lokal ini cenderung menawarkan manfaat kinerja jika dibandingkan dengan Amazon Elastic Block Store, volume EBS bukanlah pilihan saat Anda memilih jenis instans ini di OpenSearch Layanan. Jika Anda lebih suka penyimpanan EBS, gunakan tipe instans lain, seperti `r6.12xlarge.search`.

Ukuran dan jumlah serpihan

OpenSearch Pedoman umum adalah tidak melebihi 50 GB per pecahan. Mengingat jumlah serpihan yang diperlukan untuk mengakomodasi domain besar dan sumber daya yang tersedia untuk instans `i3.16xlarge.search`, kami merekomendasikan ukuran serpihan 100 GB.

Misalnya, jika Anda memiliki 450 TB sumber data dan ingin satu replika, persyaratan minimum penyimpanan Anda lebih dekat dengan 450 TB * 2 * 1,1/0,95 = 1,04 PB. Untuk penjelasan tentang perhitungan ini, lihat the section called “Menghitung persyaratan penyimpanan”. Meskipun 1,04 PB/15 TB = 70 instans, Anda dapat memilih 90 instans `i3.16xlarge.search` atau lebih untuk memberikan diri Anda jaring pengaman penyimpanan, menangani kegagalan simpul, dan

memperhitungkan beberapa varians dalam jumlah data dari waktu ke waktu. Setiap instans menambahkan 20 GiB lainnya untuk kebutuhan penyimpanan minimum Anda, tetapi untuk disk seukuran ini, 20 GiB tersebut hampir dapat diabaikan.

Mengontrol jumlah pecahan itu rumit. OpenSearch pengguna sering memutar indeks setiap hari dan menyimpan data selama satu atau dua minggu. Dalam situasi ini, Anda mungkin merasa berguna untuk membedakan antara serpihan “aktif” dan “tidak aktif”. Serpihan aktif, baik, secara aktif ditulis atau dibaca. Serpihan tidak aktif mungkin melayani beberapa permintaan baca, tetapi sebagian besar siaga. Secara umum, Anda harus menyimpan sejumlah serpihan aktif di bawah beberapa ribu. Karena jumlah serpihan aktif mendekati 10.000, risiko performa dan stabilitas yang cukup besar muncul.

Untuk menghitung jumlah serpihan primer, gunakan rumus ini: 450.000 GB * 1,1/100 GB per serpihan = 4.950 serpihan. Menggandakan jumlah tersebut untuk memperhitungkan replika adalah 9.900 serpihan, yang merupakan perhatian utama jika semua serpihan aktif. Tetapi jika Anda memutar indeks dan hanya 1/7 atau $^{1/14}$ dari pecahan yang aktif pada hari tertentu (masing-masing 1.414 atau 707 pecahan), cluster mungkin berfungsi dengan baik. Seperti biasa, langkah terpenting dalam menentukan ukuran dan mengonfigurasi domain Anda adalah melakukan pengujian klien yang representatif menggunakan kumpulan data yang realistis.

Node master khusus di OpenSearch Layanan Amazon

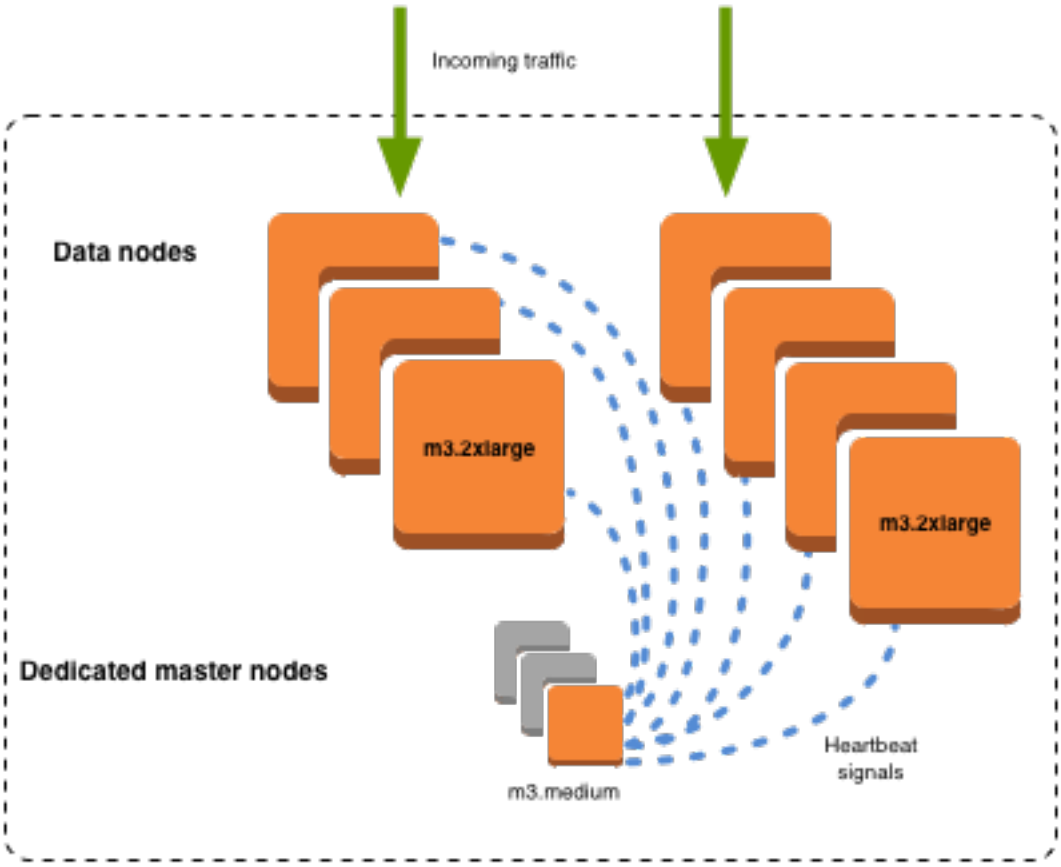
Amazon OpenSearch Service menggunakan node master khusus untuk meningkatkan stabilitas klaster. Sebuah simpul utama khusus melakukan tugas manajemen klaster, tetapi tidak menyimpan data atau menanggapi permintaan pengunggahan data. Pembongkaran tugas manajemen klaster ini meningkatkan stabilitas domain Anda. Sama seperti semua jenis simpul lainnya, Anda membayar tarif per jam untuk setiap simpul utama khusus.

Simpul utama khusus melakukan tugas manajemen klaster berikut:

- Lacak semua node di cluster.
- Lacak jumlah indeks di cluster.
- Lacak jumlah pecahan milik masing-masing indeks.
- Pertahankan informasi routing untuk node di cluster.
- Perbarui status cluster setelah perubahan status, seperti membuat indeks dan menambahkan atau menghapus node di cluster.
- Replikasi perubahan ke status cluster di semua node di cluster.

- Pantau kesehatan semua node cluster dengan mengirimkan sinyal detak jantung, sinyal periodik yang memantau ketersediaan node data dalam cluster.

Ilustrasi berikut menunjukkan domain OpenSearch Layanan dengan 10 instance. Tujuh dari instans adalah simpul data dan tiga adalah simpul utama khusus. Hanya satu dari node master khusus yang aktif. Dua node master khusus abu-abu menunggu sebagai cadangan jika node master khusus aktif gagal. Semua permintaan unggah data dilayani oleh tujuh simpul data, dan semua tugas manajemen klaster diturunkan ke simpul utama khusus yang aktif.



Memilih jumlah node master khusus

Kami menyarankan Anda menggunakan Multi-AZ dengan Standby, yang menambahkan tiga node master khusus untuk setiap domain OpenSearch Layanan produksi. Jika Anda menerapkan dengan Multi-AZ tanpa Standby atau Single-AZ, kami tetap merekomendasikan tiga node master khusus. Jangan pernah memilih jumlah simpul utama khusus yang genap. Pertimbangkan hal berikut saat memilih jumlah simpul utama khusus:

- Satu node master khusus secara eksplisit dilarang oleh OpenSearch Layanan karena Anda tidak memiliki cadangan jika terjadi kegagalan. Anda menerima pengecualian validasi jika Anda mencoba untuk membuat domain dengan hanya satu simpul utama khusus.
- Jika Anda memiliki dua node master khusus, cluster Anda tidak memiliki kuorum node yang diperlukan untuk memilih node master baru jika terjadi kegagalan.

  Kuorum adalah jumlah node master khusus/2 + 1 (dibulatkan ke bawah ke bilangan bulat terdekat). Dalam hal ini, 2 / 2 + 1 = 2. Karena satu simpul utama khusus telah gagal dan hanya satu cadangan ada, klaster tidak memiliki kuorum dan tidak dapat memilih utama baru.
- Tiga simpul utama khusus, nomor yang disarankan, menyediakan dua simpul cadangan dalam hal kegagalan simpul master dan kuorum yang diperlukan (2) untuk memilih utama baru.
- Empat node master khusus tidak lebih baik dari tiga dan dapat menyebabkan masalah jika Anda menggunakan beberapa Availability Zone.
  - Jika salah satu simpul utama gagal, Anda memiliki kuorum (3) untuk memilih utama baru. Jika dua simpul gagal, Anda kehilangan kuorum tersebut, seperti yang Anda lakukan dengan tiga simpul utama khusus.
  - Dalam konfigurasi Availability Zone tiga, dua AZ memiliki satu simpul utama khusus, dan satu AZ memiliki dua. Jika AZ mengalami gangguan, dua AZ yang tersisa tidak memiliki kuorum yang diperlukan (3) untuk memilih master baru.
- Memiliki lima simpul utama khusus berfungsi serta tiga dan memungkinkan Anda kehilangan dua simpul sambil mempertahankan kuorum. Tetapi karena hanya satu node master khusus yang aktif pada waktu tertentu, konfigurasi ini berarti Anda membayar empat node idle. Banyak pengguna menganggap tingkat perlindungan failover ini berlebihan.

Jika sebuah cluster memiliki jumlah node yang memenuhi syarat master genap, OpenSearch dan Elasticsearch versi 7. x dan kemudian abaikan satu node sehingga konfigurasi voting selalu berupa angka ganjil. Dalam kasus ini, empat simpul utama khusus pada dasarnya setara dengan tiga (dan dua banding satu).

> **Note**
>
> Jika klaster Anda tidak memiliki kuorum yang diperlukan untuk memilih simpul utama baru, menulis dan membaca permintaan untuk klaster keduanya gagal. Perilaku ini berbeda dari OpenSearch default.

Memilih jenis instance untuk node master khusus

Meskipun node master khusus tidak memproses permintaan pencarian dan kueri, ukurannya sangat berkorelasi dengan ukuran instans dan jumlah instance, indeks, dan pecahan yang dapat mereka kelola. Untuk cluster produksi, kami merekomendasikan, setidaknya, jenis instance berikut untuk node master khusus.

Rekomendasi ini didasarkan pada beban kerja yang khas dan dapat bervariasi berdasarkan kebutuhan Anda. Klaster dengan banyak serpihan atau pemetaan bidang bisa mendapatkan keuntungan dari tipe instans yang lebih besar. Memantau metrik simpul utama khusus untuk melihat apakah Anda perlu menggunakan tipe instans yang lebih besar.

Jumlah instans	Ukuran RAM simpul master	Jumlah pecahan maksimum yang didukung	Jenis instans master khusus minimum yang disarankan
1–10	8 GiB	10K	`m5.large.search` atau `m6g.large.search`
11—30	16 GiB	30K	`c5.2xlarge.search` atau `c6g.2xlarge.search`
31—75	32 GiB	40K	`r5.xlarge.search` atau `r6g.xlarge.search`
76 — 125	64 GiB	75K	`r5.2xlarge.search` atau `r6g.2xlarge.search`
126 — 200	128 GiB	75K	`r5.4xlarge.search`

Jumlah instans	Ukuran RAM simpul master	Jumlah pecahan maksimum yang didukung	Jenis instans master khusus minimum yang disarankan
			atau `r6g.4xlarge.search`

- Untuk informasi tentang bagaimana perubahan konfigurasi tertentu dapat mempengaruhi simpul utama khusus, lihat the section called “Perubahan konfigurasi”.
- Untuk klarifikasi tentang batas jumlah instans, lihat Domain OpenSearch layanan dan kuota instance.
- Untuk informasi selengkapnya tentang jenis instans tertentu, termasuk vCPU, memori, dan harga, lihat harga OpenSearch Layanan Amazon.

CloudWatch Alarm yang disarankan untuk Layanan Amazon OpenSearch

CloudWatch alarm melakukan tindakan ketika CloudWatch metrik melebihi nilai yang ditentukan untuk beberapa waktu. Misalnya, Anda mungkin AWS ingin mengirim email jika status kesehatan klaster Anda lebih dari satu menit. `red` Bagian ini mencakup beberapa alarm yang direkomendasikan untuk OpenSearch Layanan Amazon dan cara menanggapinya.

Anda dapat secara otomatis menyebarkan alarm ini menggunakan. AWS CloudFormationUntuk tumpukan sampel, lihat GitHubrepositori terkait.

> **Note**
>
> Jika Anda menerapkan CloudFormation tumpukan, `KMSKeyInaccessible` alarm `KMSKeyError` dan akan ada dalam `Insufficient Data` status karena metrik ini hanya muncul jika domain mengalami masalah dengan kunci enkripsi.

Untuk informasi selengkapnya tentang mengonfigurasi alarm, lihat Membuat CloudWatchAlarm Amazon di Panduan Pengguna Amazon CloudWatch .

Alarm	Masalah
Maksimum `ClusterStatus.red` adalah >= 1 untuk 1 menit, 1 kali berturut-turut	Setidaknya satu serpihan utama dan replika yang tidak dialokasikan untuk simpul. Lihat the section called “Status klaster merah”.
`ClusterStatus.yellow` maksimum adalah> = 1 selama 1 menit, 5 kali berturut-turut	Setidaknya satu serpihan replika tidak dialokasikan ke simpul. Lihat the section called “Status klaster kuning”.
Minimum `FreeStorageSpace` adalah <= 20480 selama 1 menit, 1 kali berturut-turut	Sebuah simpul di klaster Anda turun ke 20 GiB ruang penyimpanan gratis. Lihat the section called “Kurangnya ruang penyimpanan yang tersedia”. Nilai ini berdasarkan MiB, jadi bukan 20480, sebaiknya atur ke 25% dari ruang penyimpanan untuk setiap simpul.
`ClusterIndexWritesBlocked` adalah >= 1 untuk 5 menit, 1 kali berturut-turut	Klaster Anda memblokir permintaan tulis. Lihat the section called “ClusterBlockException”.
Minimum `Nodes` adalah < x selama 1 hari, 1 kali berturut-turut	x adalah jumlah simpul dalam klaster Anda. Alarm ini menunjukkan bahwa setidaknya satu simpul di klaster Anda telah tidak terjangkau untuk satu hari. Lihat the section called “Simpul klaster yang gagal”.
Maksimum `AutomatedSnapshotFailure` adalah >= 1 untuk 1 menit, 1 kali berturut-turut	Sebuah snapshot otomatis gagal. Kegagalan ini sering merupakan hasil dari status kesehatan klaster merah. Lihat the section called “Status klaster merah”. Untuk ringkasan semua snapshot otomatis dan beberapa informasi tentang kegagalan, cobalah salah satu permintaan berikut:

Alarm	Masalah
	`GET domain_endpoint /_snapshot/cs-automated/_all` `GET domain_endpoint /_snapshot/cs-automated-enc/_all`
Maksimum `CPUUtilization` atau `WarmCPUUtilization` adalah >= 80% untuk 15 menit, 3 kali berturut-turut	Pemanfaatan CPU 100% kadang-kadang dapat terjadi, tetapi penggunaan tinggi yang berkelanjutan bermasalah. Pertimbangkan untuk menggunakan jenis instans yang lebih besar atau menambahkan instans.
`JVMMemoryPressure` maksimum adalah> = 95% selama 1 menit, 3 kali berturut-turut	Klaster bisa mengalami kesalahan kehabisan memori jika penggunaan meningkat. Pertimbangkan penskalaan secara vertikal. OpenSearch Layanan menggunakan setengah dari RAM instance untuk heap Java, hingga ukuran heap 32 GiB. Anda dapat menskalakan instans secara vertikal hingga 64 GiB RAM, di mana Anda dapat menskalakan secara horizontal dengan menambahkan instans.
`OldGenJVMMemoryPressure` maksimum adalah> = 80% selama 1 menit, 3 kali berturut-turut	
Maksimum `MasterCPUUtilization` adalah > >= 50% untuk 15 menit, 3 kali berturut-turut	Pertimbangkan untuk menggunakan tipe instans yang lebih besar untuk simpul utama khusus. Karena peran mereka dalam stabilitas klaster dan deployment biru/hijau, simpul utama khusus harus memiliki penggunaan CPU yang lebih rendah dari simpul data.

Alarm	Masalah
`MasterJVM MemoryPre ssure` maksimum adalah> = 95% selama 1 menit, 3 kali berturut-turut	
`MasterOld GenJVMMem oryPressu re` maksimum adalah> = 80% selama 1 menit, 3 kali berturut-turut	
`KMSKeyError` adalah > = 1 untuk 1 menit, 1 kali berturut-turut	Kunci AWS KMS enkripsi yang digunakan untuk mengenkripsi data saat istirahat di domain Anda dinonaktifkan. Aktifkan kembali untuk mengembalikan operasi normal. Untuk informasi selengkapnya, lihat the section called “Enkripsi diam”.
`KMSKeyIna ccessible` adalah > = 1 untuk 1 menit, 1 kali berturut-turut	Kunci AWS KMS enkripsi yang digunakan untuk mengenkripsi data saat istirahat di domain Anda telah dihapus atau telah mencabut hibahnya ke Layanan. OpenSearch Anda tidak dapat memulihkan domain yang berada dalam keadaan ini. Namun, jika Anda memiliki snapshot manual, Anda dapat menggunakannya untuk bermigrasi ke domain baru. Untuk mempelajari selengkapnya, lihat the section called “Enkripsi diam”.
`shards.ac tive` adalah> = 30000 selama 1 menit, 1 waktu berturut-turut	Jumlah total pecahan primer dan replika aktif lebih dari 30.000. Anda mungkin memutar indeks Anda terlalu sering. Pertimbangkan untuk menggunakan ISM untuk menghapus indeks setelah mencapai usia tertentu.

Alarm	Masalah
`5xx` alarm >= 10% dari `OpenSearchRequests`	Satu atau beberapa node data mungkin kelebihan beban, atau permintaan gagal diselesaikan dalam periode batas waktu idle. Pertimbangkan untuk beralih ke jenis instance yang lebih besar atau menambahkan lebih banyak node ke cluster. Konfirmasikan bahwa Anda mengikuti praktik terbaik untuk arsitektur shard dan cluster.
`MasterReachableFromNode` maksimum < 1 selama 5 menit, 1 kali berturut-turut	Alarm ini menunjukkan bahwa node master berhenti atau tidak dapat dijangkau. Kegagalan ini biasanya merupakan hasil dari masalah konektivitas jaringan atau masalah AWS ketergantungan.
`ThreadpoolWriteQueue` Rata-rata adalah >= 100 selama 1 menit, 1 waktu berturut-turut	Cluster mengalami konkurensi pengindeksan tinggi. Meninjau dan mengontrol permintaan pengindeksan, atau meningkatkan sumber daya cluster.
`ThreadpoolSearchQueue` Rata-rata adalah> = 500 selama 1 menit, 1 waktu berturut-turut	Cluster mengalami konkurensi pencarian yang tinggi. Pertimbangkan untuk menskalakan klaster Anda. Anda juga dapat meningkatkan ukuran antrian pencarian, tetapi meningkatkannya secara berlebihan dapat menyebabkan kesalahan memori.
`ThreadpoolSearchQueue` maksimum adalah> = 5000 selama 1 menit, 1 kali berturut-turut	

Alarm	Masalah
Kenaikan `ThreadpoolSearchRejected` SUM adalah >=1 {ekspresi matematika DIFF ()} selama 1 menit, 1 waktu berturut-turut	Alarm ini memberi tahu Anda tentang masalah domain yang mungkin memengaruhi kinerja dan stabilitas.
Kenaikan `ThreadpoolWriteRejected` SUM adalah >=1 {ekspresi matematika DIFF ()} selama 1 menit, 1 waktu berturut-turut	

> **Note**
>
> Jika Anda hanya ingin melihat metrik, lihatthe section called “Memantau metrik klaster”.

Alarm lain yang mungkin Anda pertimbangkan

Pertimbangkan untuk mengonfigurasi alarm berikut tergantung pada fitur OpenSearch Layanan yang sering Anda gunakan.

Alarm	Isu
`WarmFreeStorageSpace` minimum <= 10240 selama 1 menit, 1 kali berturut-turut	Sebuah UltraWarm node di cluster Anda turun ke 10 GiB ruang penyimpanan gratis. Lihat the section called “Kurangnya ruang penyimpanan yang tersedia”. Nilai ini ada di MiB, jadi daripada 10240, kami sarankan untuk mengaturnya ke 10% dari ruang penyimpanan untuk setiap node. UltraWarm

Alarm	Isu
`HotToWarm Migration QueueSize` adalah> = 20 selama 1 menit, 3 kali berturut-turut	Sejumlah besar indeks secara bersamaan bergerak dari panas ke UltraWarm penyimpanan. Pertimbangkan untuk menskalakan klaster Anda.
`HotToWarm Migration SuccessLa tency` adalah> = 1 hari, 1 kali berturut-turut	Konfigurasikan alarm ini sehingga Anda diberi tahu jika latensi `HotToWarmMigrationSuccessCount` x lebih besar dari 24 jam jika Anda mencoba memutar indeks harian.
`WarmJVMMe moryPress ure` maksimum adalah> = 95% selama 1 menit, 3 kali berturut-turut	Klaster bisa mengalami kesalahan kehabisan memori jika penggunaa n meningkat. Pertimbangkan penskalaan secara vertikal. OpenSearch Layanan menggunakan setengah dari RAM instance untuk heap Java, hingga ukuran heap 32 GiB. Anda dapat menskalakan instans secara vertikal hingga 64 GiB RAM, di mana Anda dapat menskalakan secara horizontal dengan menambahkan instans.
`WarmOldGe nJVMMemor yPressure` maksimum adalah> = 80% selama 1 menit, 3 kali berturut-turut	
`WarmToCol dMigratio nQueueSiz e` adalah> = 20 selama 1 menit, 3 kali berturut-turut	Sejumlah besar indeks secara bersamaan berpindah dari UltraWarm ke cold storage. Pertimbangkan untuk menskalakan klaster Anda.

Alarm	Isu
`HotToWarm Migration FailureCount` adalah > = 1 untuk 1 menit, 1 kali berturut-turut	Migrasi mungkin gagal selama snapshot, relokasi pecahan, atau penggabungan paksa. Kegagalan selama snapshot atau relokasi serpihan biasanya karena kegagalan simpul atau masalah konektivi tas S3. Kurangnya ruang disk biasanya menjadi penyebab kegagalan penggabungan paksa.
`WarmToCol dMigratio nFailureCount` adalah > = 1 untuk 1 menit, 1 kali berturut-turut	Migrasi biasanya gagal ketika upaya untuk memigrasikan metadata indeks ke penyimpanan dingin gagal. Kegagalan juga dapat terjadi ketika status cluster indeks hangat sedang dihapus.
`WarmToCol dMigratio nLatency` adalah> = 1 hari, 1 kali berturut-turut	Konfigurasikan alarm ini sehingga Anda diberi tahu jika latensi `WarmToColdMigrationSuccessCount` x lebih besar dari 24 jam jika Anda mencoba memutar indeks harian.
`AlertingDegraded` adalah > = 1 untuk 1 menit, 1 kali berturut-turut	Entah indeks peringatan berwarna merah, atau satu atau lebih node tidak sesuai jadwal.
`ADPluginU nhealthy` adalah > = 1 untuk 1 menit, 1 kali berturut-turut	Plugin deteksi anomali tidak berfungsi dengan baik, baik karena tingkat kegagalan yang tinggi atau karena salah satu indeks yang digunakan berwarna merah.

Alarm	Isu
`AsynchronousSearchFailureRate` adalah > = 1 untuk 1 menit, 1 kali berturut-turut	Setidaknya satu pencarian asinkron gagal di menit terakhir, yang kemungkinan berarti node koordinator gagal. Siklus hidup permintaan pencarian asinkron dikelola hanya pada node koordinator, jadi jika koordinator turun, permintaan gagal.
`AsynchronousSearchStoreHealth` adalah > = 1 untuk 1 menit, 1 kali berturut-turut	Kesehatan penyimpanan respons pencarian asinkron dalam indeks bertahan berwarna merah. Anda mungkin menyimpan respons asinkron besar, yang dapat mengacaukan klaster. Cobalah untuk membatasi respons pencarian asinkron Anda hingga 10 MB atau kurang.
`SQLUnhealthy` adalah> = 1 selama 1 menit, 3 kali berturut-turut	Plugin SQL mengembalikan kode respons 5 xx atau meneruskan kueri DSL yang tidak valid ke. OpenSearch Memecahkan masalah permintaan yang klien Anda buat untuk plugin.
`LTRStatus.red` adalah > = 1 untuk 1 menit, 1 kali berturut-turut	Setidaknya salah satu indeks yang diperlukan untuk menjalankan plugin Learning to Rank memiliki pecahan primer yang hilang dan tidak berfungsi.

Referensi umum untuk Amazon OpenSearch Service

Amazon OpenSearch Service mendukung berbagai instans, operasi, plugin, dan sumber daya lainnya.

Topik

Jenis instans yang didukung di Amazon OpenSearch Service

Amazon OpenSearch Service mendukung jenis instans berikut. Tidak semua Wilayah mendukung semua tipe instans. Untuk detail ketersediaan, lihat harga OpenSearch Layanan Amazon.

Untuk informasi tentang tipe instans yang sesuai untuk kasus penggunaan Anda, lihat the section called “Mengukur domain”, the section called “Kuota ukuran volume EBS”, dan the section called “Kuota jaringan”.

Jenis instance generasi saat ini

Untuk performa terbaik, sebaiknya gunakan jenis instans berikut saat membuat domain OpenSearch Layanan baru.

Jenis instans	Instans	Pembatasan
ATAU1	`or1.medium.search` `or1.large.search`	• Jenis instans OR1 membutuhkan OpenSearch 2.11 atau yang lebih baru. • Instance OR1 hanya kompatibel dengan node master tipe instance Graviton lainnya (C6g, m6g, R6g).

Jenis instans	Instans	Pembatasan
	or1.xlarge.search or1.2xlarge.search or1.4xlarge.search or1.8xlarge.search or1.12xlarge.search or1.16xlarge.search	

Jenis instans	Instans	Pembatasan

Im4gn	`im4gn.large.search`
	`im4gn.xlarge.search`
	`im4gn.2xlarge.search`
	`im4gn.4xlarge.search`
	`im4gn.8xlarge.search`
	`im4gn.16xlarge.search`
		• Tipe instans iM4gn memerlukan Elasticsearch 7.9 atau yang lebih baru atau versi apa pun OpenSearch, dan tidak mendukung volume penyimpanan EBS.
		• Instans IM4gn hanya kompatibel dengan jenis instans Graviton lainnya (C6g, m6g, R6g, R6gd). Anda tidak dapat menggabungkan instans Graviton dan non-Graviton dalam klaster yang sama.

Jenis instans	Instans	Pembatasan

C5	`c5.large.search`
	`c5.xlarge.search`
	`c5.2xlarge.search`
	`c5.4xlarge.search`
	`c5.9xlarge.search`
	`c5.18xlarge.search`
		Jenis instans C5 memerlukan Elasticsearch 5.1 atau yang lebih baru atau versi apa pun. OpenSearch

Jenis instans	Instans	Pembatasan

C6g	`c6g.large.search`
	`c6g.xlarge.search`
	`c6g.2xlarge.search`
	`c6g.4xlarge.search`
	`c6g.8xlarge.search`
	`c6g.12xlarge.search`
		• Jenis instans C6g memerlukan Elasticsearch 7.9 atau yang lebih baru atau versi apa pun. OpenSearch
		• Instans C6g hanya kompatibel dengan jenis instans Graviton lainnya (iM4gn, M6g, R6g, R6gd). Anda tidak dapat menggabungkan instans Graviton dan non-Graviton dalam klaster yang sama.

Jenis instans	Instans	Pembatasan

I3	`i3.large.search`
	`i3.xlarge.search`
	`i3.2xlarge.search`
	`i3.4xlarge.search`
	`i3.8xlarge.search`
	`i3.16xlarge.search`
		Jenis instans I3 memerlukan Elasticsearch 5.1 atau yang lebih baru atau versi apa pun OpenSearch, dan tidak mendukung volume penyimpanan EBS.

M5	`m5.large.search`
	`m5.xlarge.search`
	`m5.2xlarge.search`
	`m5.4xlarge.search`
	`m5.12xlarge.search`
		Jenis instans M5 memerlukan Elasticsearch 5.1 atau yang lebih baru atau versi apa pun. OpenSearch

Jenis instans	Instans	Pembatasan

M6g	`m6g.large.search`
	`m6g.xlarge.search`
	`m6g.2xlarge.search`
	`m6g.4xlarge.search`
	`m6g.8xlarge.search`
	`m6g.12xlarge.search`
		• Jenis instans M6g memerlukan Elasticsearch 7.9 atau yang lebih baru atau versi apa pun. OpenSearch
		• Instans M6g hanya kompatibel dengan jenis instans Graviton lainnya (IM4gn, C6g, R6g, R6gd). Anda tidak dapat menggabungkan instans Graviton dan non-Graviton dalam klaster yang sama.

Jenis instans	Instans	Pembatasan

R5	`r5.large.search`
	`r5.xlarge.search`
	`r5.2xlarge.search`
	`r5.4xlarge.search`
	`r5.12xlarge.search`
		Jenis instans R5 memerlukan Elasticsearch 5.1 atau yang lebih baru atau versi apa pun. OpenSearch

Jenis instans	Instans	Pembatasan

R6g	`r6g.large.search`
	`r6g.xlarge.search`
	`r6g.2xlarge.search`
	`r6g.4xlarge.search`
	`r6g.8xlarge.search`
	`r6g.12xlarge.search`
		• Jenis instans R6g memerlukan Elasticsearch 7.9 atau yang lebih baru atau versi apa pun. OpenSearch
		• Instans R6g hanya kompatibel dengan jenis instans Graviton lainnya (iM4gn, C6g, M6g, R6gd). Anda tidak dapat menggabungkan instans Graviton dan non-Graviton dalam klaster yang sama.

Jenis instans	Instans	Pembatasan

R6gd	`r6gd.large.search`
	`r6gd.xlarge.search`
	`r6gd.2xlarge.search`
	`r6gd.4xlarge.search`
	`r6gd.8xlarge.search`
	`r6gd.12xlarge.search`
	`r6gd.16xlarge.search`
		• Jenis instans R6gd memerlukan Elasticsearch 7.9 atau yang lebih baru atau versi apa pun OpenSearch, dan tidak mendukung volume penyimpanan EBS.
		• Instans R6gd hanya kompatibel dengan jenis instans Graviton lainnya (iM4gn, C6g, M6g, R6g). Anda tidak dapat menggabungkan instans Graviton dan non-Graviton dalam klaster yang sama.

Jenis instans	Instans	Pembatasan
T3	`t3.small.search` `t3.medium.search`	• Jenis instans T3 memerlukan Elasticsearch 5.6 atau yang lebih baru atau versi apa pun. OpenSearch • Anda dapat menggunakan tipe instans T3 hanya jika domain Anda disediakan tanpa siaga. Untuk informasi selengkapnya, lihat the section called “Multi-AZ tanpa Siaga”. • Anda dapat menggunakan tipe instans T3 hanya jika jumlah instans untuk domain Anda adalah 10 atau kurang. • Jenis instans T3 tidak mendukung UltraWarm penyimpanan, penyimpanan dingin, atau Auto-Tune.

Tipe instans generasi sebelumnya

OpenSearch Layanan menawarkan jenis instance generasi sebelumnya untuk pengguna yang telah mengoptimalkan aplikasi mereka di sekitar mereka dan belum meningkatkan. Kami mendorong Anda untuk menggunakan tipe instans generasi saat ini agar bisa mendapatkan performa terbaik, tetapi kami terus mendukung tipe instans generasi sebelumnya berikut.

Jenis instans	Instans	Pembatasan

C4	`c4.large.search`
	`c4.xlarge.search`
	`c4.2xlarge.search`
	`c4.4xlarge.search`	

Jenis instans	Instans	Pembatasan
	`c4.8xlarge.search`	
I2	`i2.xlarge.search` `i2.2xlarge.search`	

M3	`m3.medium.search`
	`m3.large.search`
	`m3.xlarge.search`
	`m3.2xlarge.search`
		• Tipe instans M3 tidak mendukung enkripsi data saat tidak digunakan, kontrol akses detail, atau pencarian lintas klaster.
		• Jenis instans M3 memiliki batasan tambahan berdasarkan OpenSearch versi. Untuk mempelajari informasi lebih lanjut, lihat the section called “Tipe instans M3 tidak valid”.

M4	`m4.large.search`
	`m4.xlarge.search`
	`m4.2xlarge.search`
	`m4.4xlarge.search`
	`m4.10xlarge.search`	

Jenis instans	Instans	Pembatasan

R3	`r3.large.search`
	`r3.xlarge.search`
	`r3.2xlarge.search`
	`r3.4xlarge.search`
	`r3.8xlarge.search`
		Tipe instans R3 tidak mendukung enkripsi data saat tidak digunakan atau kontrol akses detail.

R4	`r4.large.search`
	`r4.xlarge.search`
	`r4.2xlarge.search`
	`r4.4xlarge.search`
	`r4.8xlarge.search`
	`r4.16xlarge.search`	

Jenis instans	Instans	Pembatasan

T2	`t2.micro.search`
	`t2.small.search`
	`t2.medium.search`
		• Anda dapat menggunakan tipe instans T2 hanya jika jumlah instans untuk domain Anda adalah 10 atau lebih sedikit.
		• Tipe instans `t2.micro.search` hanya mendukung Elasticsearch 1.5 dan 2.3.
		• Jenis instans T2 tidak mendukung enkripsi data saat istirahat, kontrol akses berbutir halus, UltraWarm penyimpanan, penyimpanan dingin, pencarian lintas cluster, atau Auto-Tune.

> **Tip**
>
> Kami sering merekomendasikan tipe instans berbeda untuk simpul master khusus dan simpul data.

Fitur berdasarkan versi mesin di Amazon OpenSearch Service

Banyak fitur OpenSearch Layanan memiliki persyaratan OpenSearch versi minimum atau persyaratan versi Elasticsearch OSS lama. Jika Anda memenuhi versi minimum untuk suatu fitur, tetapi fitur tersebut tidak tersedia di domain Anda, perbarui perangkat lunak layanan domain Anda.

Fitur	OpenSearch Versi minimum yang diperlukan	Versi Elasticsearch minimum yang diperlukan
Dukungan VPC	1.0	1.0
Memerlukan HTTPS untuk semua lalu lintas ke domain		

Fitur	OpenSearch Versi minimum yang diperlukan	Versi Elasticsearch minimum yang diperlukan
Dukungan Multi-AZ		
Simpul utama khusus		
Paket kustom		
Titik akhir kustom		
Penerbitan log lambat		
Kesalahan penerbitan log	1.0	5.1
Enkripsi data saat tidak digunakan		
Otentikasi Cognito untuk Dasbor OpenSearch		
Peningkatan di tempat		

Fitur	OpenSearch Versi minimum yang diperlukan	Versi Elasticsearch minimum yang diperlukan
Dukungan kurator	Tidak termasuk	5.1
Snapshot otomatis setiap jam	1.0	5.3
ode-to-node Enkripsi N	1.0	6.0
Dukungan klien REST level-tinggi Java		
Permintaan HTTP dan kompresi respons		
Memberi peringatan	1.0	6.2
SQL	1.0	6.5
Pencarian lintas klaster	1.0	6.7
Kontrol akses detail		
Otentikasi SAM untuk Dasbor OpenSearch		

Fitur	OpenSearch Versi minimum yang diperlukan	Versi Elasticsearch minimum yang diperlukan
Auto-Tune		
Indeks ulang jarak jauh		
UltraWarm	1.0	6.8
Manajemen State Indeks		
k-NN berdasarkan jarak Euclidean	1.0	7.1
Deteksi Anomali	1.0	7.4
k-NN oleh kesamaan kosinus	1.0	7.7
Belajar untuk Peringkat		
Bahasa pemrosesan yang disalurkan	1.0	7.9
OpenSearch Laporan dasbor		

Fitur	OpenSearch Versi minimum yang diperlukan	Versi Elasticsearch minimum yang diperlukan
OpenSearch Analisis Pelacakan Dasbor		
Instans Graviton berbasis ARM		
Penyimpanan dingin		
Jarak Hamming, jarak L1 Norm, dan penulisan Painless untuk k-NN	1.0	7.10
Pencarian asinkron		
Indeks berubah	1.0	Tidak termasuk
Replikasi lintas cluster	1.1	7.10
ML Commons	1.3	Tidak termasuk
Notifikasi	2.3	Tidak termasuk

Fitur	OpenSearch Versi minimum yang diperlukan	Versi Elasticsearch minimum yang diperlukan
Pencarian titik waktu	2.5	Tidak termasuk
Cari saluran pipa	2.9	Tidak termasuk
Konektor pembelajaran mesin	2.9	Tidak termasuk
Pencarian semantik multimodal	2.11	Tidak termasuk
Sumber data kueri langsung untuk Amazon S3	2.11	Tidak termasuk

Untuk informasi tentang plugin, yang memungkinkan beberapa fitur ini dan fungsionalitas tambahan, lihat the section called “Plugin berdasarkan versi mesin”. Untuk informasi tentang OpenSearch API untuk setiap versi, lihatthe section called “Operasi yang didukung”.

Plugin berdasarkan versi mesin di Amazon Service OpenSearch

Domain OpenSearch Layanan Amazon dikemas dengan plugin dari komunitas. OpenSearch Layanan ini secara otomatis menyebarkan dan mengelola plugin untuk Anda, tetapi menyebarkan plugin yang berbeda tergantung pada versi OpenSearch atau lama Elasticsearch OSS yang Anda pilih untuk domain Anda.

Tabel berikut mencantumkan plugin berdasarkan OpenSearch versi, serta versi yang kompatibel dari Elasticsearch OSS lama. Ini hanya mencakup plugin yang mungkin berinteraksi dengan Anda — itu tidak komprehensif. OpenSearch Layanan menggunakan plugin tambahan untuk mengaktifkan fungsionalitas layanan inti, seperti plugin S3 Repository untuk snapshot dan plugin

OpenSearchPerformance Analyzer untuk optimasi dan pemantauan. Untuk daftar lengkap semua plugin yang berjalan di domain Anda, buat permintaan berikut:

```
GET _cat/plugins?v
```

Plugin	OpenSearch Versi minimum yang diperlukan	Versi Elasticsearch minimum yang diperlukan
Analisis ICU	1.0	Termasuk pada semua domain
Analisis Bahasa Jepang (Kuromoji)		
Analisis Fonetik	1.0	2.3
Analisis Korea Seunjeon	1.0	5.1
Analisis Bahasa Mandarin Cerdas		
Analisis Stempel Polandia		
Prosesor Lampiran Serap		
Prosesor Agen		

Plugin	OpenSearch Versi minimum yang diperlukan	Versi Elasticsearch minimum yang diperlukan
Pengguna Serap		
Pemeta Murmur3		
Ukuran Pemeta	1.0	5.3
Analisis Bahasa Ukraina		
OpenSearch mengingatkan	1.0	6.2
OpenSearch SQL	1.0	6.5
OpenSearch keamanan	1.0	6.7
OpenSearch Indeks Manajemen Negara	1.0	6.8
OpenSearch K-nn	1.0	7.1
OpenSearch deteksi anomali	1.0	7.4

Plugin	OpenSearch Versi minimum yang diperlukan	Versi Elasticsearch minimum yang diperlukan
Analisis IK (Mandarin)	1.0	7.7
Analisis Vietnam		
Analisis Thailand		
Belajar Rank		
OpenSearch pencarian asinkron	1.0	7.10
OpenSearch replikasi lintas-cluster	1.1	7.10
OpenSearch observabilitas	1.2	Tidak didukung
Analisis Nori	1.3	Tidak didukung
Analisis Pinyin	1.3	Tidak didukung
STConvert	1.3	Tidak didukung
Analisis Sudachi	1.3	Tidak didukung

Plugin	OpenSearch Versi minimum yang diperlukan	Versi Elasticsearch minimum yang diperlukan
ML Commons	1.3	Tidak didukung
OpenSearch pemberitahuan	2.3	Tidak didukung
Analisis Keamanan	2.5	Tidak didukung
Pencarian Saraf	2.9	Tidak didukung
Amazon Personalisasi Peringkat Pencarian	2.9	Tidak didukung

Plugin opsional

Selain plugin default yang sudah diinstal sebelumnya, Amazon OpenSearch Service mendukung beberapa plugin penganalisis bahasa. Plugin ini ditandai sebagai opsional pada tabel di atas. Anda dapat menggunakan AWS Management Console dan AWS CLI untuk mengaitkan plugin ke domain, memisahkan plugin dari domain, dan daftar semua plugin. Paket plugin opsional kompatibel dengan OpenSearch versi tertentu, dan hanya dapat dikaitkan dengan domain dengan versi itu.

Perhatikan bahwa untuk plugin Sudachi, ketika Anda mengasosiasikan kembali file kamus, itu tidak langsung mencerminkan domain. Kamus menyegarkan ketika penerapan biru/hijau berikutnya berjalan pada domain sebagai bagian dari perubahan konfigurasi atau pembaruan lainnya. Atau, Anda dapat membuat indeks baru, mengindeks ulang indeks yang ada ke indeks baru, dan kemudian menghapus indeks lama. Jika Anda lebih suka menggunakan pendekatan pengindeksan ulang, gunakan alias indeks sehingga tidak ada gangguan pada lalu lintas Anda.

Plugin opsional menggunakan jenis `ZIP-PLUGIN` paket. Untuk informasi selengkapnya tentang plugin opsional, lihatthe section called “Paket kustom”.

Operasi yang didukung di Amazon OpenSearch Service

OpenSearch Layanan mendukung banyak versi OpenSearch dan warisan Elasticsearch OSS. Bagian berikut menunjukkan operasi yang didukung OpenSearch Layanan untuk setiap versi.

Topik

- Elasticsearch versi 5.6
- Elasticsearch versi 5.5
- Elasticsearch versi 5.3
- Elasticsearch versi 5.1
- Elasticsearch versi 2.3
- Elasticsearch versi 1.5

Perbedaan API yang mencolok

Pengaturan dan statistik

OpenSearch Layanan hanya menerima permintaan PUT ke `_cluster/settings` API yang menggunakan formulir pengaturan “datar”. Pengaturan ini menolak permintaan yang menggunakan formulir pengaturan diperluas.

```
// Accepted
PUT _cluster/settings
{
  "persistent" : {
    "action.auto_create_index" : false
  }
}

// Rejected
PUT _cluster/settings
{
  "persistent": {
    "action": {
      "auto_create_index": false
    }
  }
}
```

Klien Java REST tingkat tinggi menggunakan formulir yang diperluas, jadi jika Anda perlu mengirim permintaan pengaturan, gunakan klien tingkat rendah.

Sebelum Elasticsearch 5.3, `_cluster/settings` API pada domain OpenSearch Layanan hanya mendukung PUT metode HTTP, bukan metode. `GET` OpenSearch dan versi Elasticsearch yang lebih baru mendukung `GET` metode ini, seperti yang ditunjukkan pada contoh berikut:

```
GET https://domain-name.region.es.amazonaws.com/_cluster/settings?pretty
```

Berikut adalah contoh kembali:

```
{
  "persistent": {
    "cluster": {
      "routing": {
        "allocation": {
          "cluster_concurrent_rebalance": "2",
          "node_concurrent_recoveries": "2",
          "disk": {
            "watermark": {
              "low": "1.35gb",
              "flood_stage": "0.45gb",
              "high": "0.9gb"
            }
          },
          "node_initial_primarirecoveries": "4"
        }
      }
    },
    "indices": {
      "recovery": {
        "max_bytper_sec": "40mb"
      }
    }
  }
}
```

Jika Anda membandingkan respons dari OpenSearch kluster sumber terbuka dan OpenSearch Layanan untuk setelan dan API statistik tertentu, Anda mungkin melihat bidang yang hilang. OpenSearch Layanan menyunting informasi tertentu yang mengekspos internal layanan, seperti jalur data sistem file dari `_nodes/stats` atau nama sistem operasi dan versi dari. `_nodes`

Kecilkan

API `_shrink` dapat menyebabkan kegagalan pembaruan, perubahan konfigurasi, dan penghapusan domain. Kami tidak menyarankan untuk menggunakannya pada domain yang menjalankan Elasticsearch versi 5.3 atau 5.1. Versi ini memiliki bug yang dapat menyebabkan kegagalan pemulihan snapshot indeks yang menyusut.

Jika Anda menggunakan `_shrink` API di Elasticsearch atau OpenSearch versi lain, buat permintaan berikut sebelum memulai operasi shrink:

```
PUT https://domain-name.region.es.amazonaws.com/source-index/_settings
{
  "settings": {
    "index.routing.allocation.require._name": "name-of-the-node-to-shrink-to",
    "index.blocks.read_only": true
  }
}
```

Kemudian buat permintaan berikut setelah menyelesaikan operasi menyusut:

```
PUT https://domain-name.region.es.amazonaws.com/source-index/_settings
{
  "settings": {
    "index.routing.allocation.require._name": null,
    "index.blocks.read_only": false
  }
}

PUT https://domain-name.region.es.amazonaws.com/shrunken-index/_settings
{
  "settings": {
    "index.routing.allocation.require._name": null,
    "index.blocks.read_only": false
  }
}
```

OpenSearch versi 2.11

Untuk OpenSearch 2.11, OpenSearch Layanan mendukung operasi berikut. Untuk informasi tentang sebagian besar operasi, lihat referensi OpenSearch REST API, atau referensi API untuk plugin tertentu.

- Semua operasi di jalur indeks (seperti */index-name* */_forcemerge* , */index-name* */update/id*, dan */index-name* */_close*)
- `/_delete_by_query` [1]
- `/_explain`
- `/_field_caps`
- `/_field_stats`
- `/_refresh`
- `/_reindex` [1]
- `/_render`
- `/_resolve/index`

- /_alias
- /_aliases
- /_all
- /_analyze
- /_bulk
- /_cat (kecuali /_cat/nodeattrs)
- /_cluster/allocation/explain
- /_cluster/health
- /_cluster/pending_tasks
- /_cluster/settings untuk beberapa properti[4]:
  - action.auto_create_index
  - action.search.shard_count.limit
  - indices.breaker.fielddata.limit
  - indices.breaker.request.limit
  - indices.breaker.total.limit
  - cluster.max_shards_per_node
- /_cluster/state
- /_cluster/stats
- /_count
- /_dashboards
- /_flush
- /_ingest/pipeline
- /_ltr
- /_mapping
- /_mget
- /_msearch
- /_mtermvectors
- /_nodes
- /_plugins/_asynchronous_search
- /_plugins/_alerting
- /_plugins/_anomaly_detection
- /_plugins/_ism
- /_plugins/_ml
- /_plugins/_notifications
- /_plugins/_ppl
- /_plugins/_security
- /_plugins/_security_analytics
- /_plugins/_sm
- /_plugins/_sql
- /_percolate
- /_rank_eval
- /_rollover
- /_scripts [3]
- /_search[2]
- /_search/pipeline
- /_search/point_in_time
- /_search profile
- /_shard_stores
- /_shrink[5]
- /_snapshot
- /_split
- /_stats
- /_status
- /_tasks
- /_template
- /_update_by_query [1]
- /_validate

1. Perubahan konfigurasi klaster mungkin mengganggu operasi ini sebelum selesai. Kami menyarankan agar Anda menggunakan operasi `/_tasks` bersama dengan operasi ini untuk memverifikasi bahwa permintaan berhasil diselesaikan.
2. HAPUS permintaan untuk `/_search/scroll` dengan badan pesan harus menentukan `"Content-Length"` di header HTTP. Kebanyakan klien menambahkan header ini secara default. Untuk menghindari masalah dengan = karakter dalam `scroll_id` nilai, gunakan badan permintaan, bukan string kueri, untuk meneruskan `scroll_id` nilai ke OpenSearch Service.
3. Untuk pertimbangan tentang menggunakan penulisan, lihat the section called “Sumber daya lain yang didukung”.
4. Lihat metode PUT. Untuk informasi selengkapnya tentang metode GET, lihat the section called “Perbedaan API yang mencolok”. Daftar ini hanya mengacu pada OpenSearch operasi generik yang didukung OpenSearch Layanan dan tidak menyertakan operasi yang didukung khusus plugin untuk deteksi anomali, ISM, dan sebagainya.
5. Lihat the section called “Kecilkan”.

OpenSearch versi 2.9

Untuk OpenSearch 2.9, OpenSearch Layanan mendukung operasi berikut. Untuk informasi tentang sebagian besar operasi, lihat referensi OpenSearch REST API, atau referensi API untuk plugin tertentu.

- Semua operasi di jalur indeks (seperti */index-name* `/_forcemerge` , */index-name* `/update/`*id*, dan */index-name* `/_close`)
- `/_alias`
- `/_aliases`
- `/_all`
- `/_analyze`
- `/_bulk`
- `/_cat` (kecuali `/_cat/nodeattrs`)
- `/_delete_by_query` [1]
- `/_explain`
- `/_field_caps`
- `/_field_stats`
- `/_flush`
- `/_ingest/pipeline`
- `/_ltr`
- `/_mapping`
- `/_mget`
- `/_msearch`
- `/_mtermvectors`
- `/_nodes`
- `/_refresh`
- `/_reindex` [1]
- `/_render`
- `/_resolve/index`
- `/_rollover`
- `/_scripts` [3]
- `/_search`[2]
- `/_search/pipeline`
- `/_search/point_in_time`
- `/_search profile`
- `/_shard_stores`

- `/_cluster/allocation/explain`
- `/_cluster/health`
- `/_cluster/pending_tasks`
- `/_cluster/settings` untuk beberapa properti[4]:
  - `action.auto_create_index`
  - `action.search.shard_count.limit`
  - `indices.breaker.fielddata.limit`
  - `indices.breaker.request.limit`
  - `indices.breaker.total.limit`
  - `cluster.max_shards_per_node`
- `/_cluster/state`
- `/_cluster/stats`
- `/_count`
- `/_dashboards`
- `/_plugins/_asynchronous_search`
- `/_plugins/_alerting`
- `/_plugins/_anomaly_detection`
- `/_plugins/_ism`
- `/_plugins/_ml`
- `/_plugins/_notifications`
- `/_plugins/_ppl`
- `/_plugins/_security`
- `/_plugins/_security_analytics`
- `/_plugins/_sm`
- `/_plugins/_sql`
- `/_percolate`
- `/_rank_eval`
- `/_shrink`[5]
- `/_snapshot`
- `/_split`
- `/_stats`
- `/_status`
- `/_tasks`
- `/_template`
- `/_update_by_query` [1]
- `/_validate`

1. Perubahan konfigurasi klaster mungkin mengganggu operasi ini sebelum selesai. Kami menyarankan agar Anda menggunakan operasi `/_tasks` bersama dengan operasi ini untuk memverifikasi bahwa permintaan berhasil diselesaikan.
2. HAPUS permintaan untuk `/_search/scroll` dengan badan pesan harus menentukan `"Content-Length"` di header HTTP. Kebanyakan klien menambahkan header ini secara default. Untuk menghindari masalah dengan = karakter dalam `scroll_id` nilai, gunakan badan permintaan, bukan string kueri, untuk meneruskan `scroll_id` nilai ke OpenSearch Service.
3. Untuk pertimbangan tentang menggunakan penulisan, lihat the section called “Sumber daya lain yang didukung”.

4. Lihat metode PUT. Untuk informasi selengkapnya tentang metode GET, lihat the section called "Perbedaan API yang mencolok". Daftar ini hanya mengacu pada OpenSearch operasi generik yang didukung OpenSearch Layanan dan tidak menyertakan operasi yang didukung khusus plugin untuk deteksi anomali, ISM, dan sebagainya.
5. Lihat the section called "Kecilkan".

OpenSearch versi 2.7

Untuk OpenSearch 2.7, OpenSearch Layanan mendukung operasi berikut. Untuk informasi tentang sebagian besar operasi, lihat referensi OpenSearch REST API, atau referensi API untuk plugin tertentu.

- Semua operasi di jalur indeks (seperti */index-name* /_forcemerge , */index-name* /update/*id*, dan */index-name* /_close)
- /_alias
- /_aliases
- /_all
- /_analyze
- /_bulk
- /_cat (kecuali /_cat/nodeattrs)
- /_cluster/allocation/explain
- /_cluster/health
- /_cluster/pending_tasks
- /_cluster/settings untuk beberapa properti[4]:
  - action.auto_create_index
- /_delete_by_query [1]
- /_explain
- /_field_caps
- /_field_stats
- /_flush
- /_ingest/pipeline
- /_ltr
- /_mapping
- /_mget
- /_msearch
- /_mtermvectors
- /_nodes
- /_plugins/_asynchronous_search
- /_plugins/_alerting
- /_plugins/_anomaly_detection
- /_plugins/_ism
- /_plugins/_ml
- /_refresh
- /_reindex [1]
- /_render
- /_resolve/index
- /_rollover
- /_scripts [3]
- /_search[2]
- /_search/point_in_time
- /_search profile
- /_shard_stores
- /_shrink[5]
- /_snapshot
- /_split
- /_stats
- /_status
- /_tasks
- /_template
- /_update_by_query [1]
- /_validate

  - action.search.shard_count.limit
  - indices.breaker.fielddata.limit
  - indices.breaker.request.limit
  - indices.breaker.total.limit
  - cluster.max_shards_per_node
- /_cluster/state
- /_cluster/stats
- /_count
- /_dashboards
- /_plugins/_notifications
- /_plugins/_ppl
- /_plugins/_security
- /_plugins/_security_analytics
- /_plugins/_sm
- /_plugins/_sql
- /_percolate
- /_rank_eval

1. Perubahan konfigurasi klaster mungkin mengganggu operasi ini sebelum selesai. Kami menyarankan agar Anda menggunakan operasi `/_tasks` bersama dengan operasi ini untuk memverifikasi bahwa permintaan berhasil diselesaikan.
2. HAPUS permintaan untuk `/_search/scroll` dengan badan pesan harus menentukan `"Content-Length"` di header HTTP. Kebanyakan klien menambahkan header ini secara default. Untuk menghindari masalah dengan = karakter dalam `scroll_id` nilai, gunakan badan permintaan, bukan string kueri, untuk meneruskan `scroll_id` nilai ke OpenSearch Service.
3. Untuk pertimbangan tentang menggunakan penulisan, lihat the section called “Sumber daya lain yang didukung”.
4. Lihat metode `PUT`. Untuk informasi selengkapnya tentang metode `GET`, lihat the section called “Perbedaan API yang mencolok”. Daftar ini hanya mengacu pada OpenSearch operasi generik yang didukung OpenSearch Layanan dan tidak menyertakan operasi yang didukung khusus plugin untuk deteksi anomali, ISM, dan sebagainya.
5. Lihat the section called “Kecilkan”.

OpenSearch versi 2.5

Untuk OpenSearch 2.5, OpenSearch Layanan mendukung operasi berikut. Untuk informasi tentang sebagian besar operasi, lihat referensi OpenSearch REST API, atau referensi API untuk plugin tertentu.

- Semua operasi di jalur indeks (seperti */index-name* /_forcemerge , */index-name* /update/*id*, dan */index-name* /_close)
- /_alias
- /_aliases
- /_all
- /_analyze
- /_bulk
- /_cat (kecuali /_cat/nodeattrs)
- /_cluster/allocation/explain
- /_cluster/health
- /_cluster/pending_tasks
- /_cluster/settings untuk beberapa properti[4]:
  - action.auto_create_index
  - action.search.shard_count.limit
  - indices.breaker.fielddata.limit
  - indices.breaker.request.limit
- /_delete_by_query [1]
- /_explain
- /_field_caps
- /_field_stats
- /_flush
- /_ingest/pipeline
- /_ltr
- /_mapping
- /_mget
- /_msearch
- /_mtermvectors
- /_nodes
- /_plugins/_asynchronous_search
- /_plugins/_alerting
- /_plugins/_anomaly_detection
- /_plugins/_ism
- /_plugins/_ml
- /_plugins/_notifications
- /_plugins/_ppl
- /_plugins/_security
- /_refresh
- /_reindex [1]
- /_render
- /_resolve/index
- /_rollover
- /_scripts [3]
- /_search[2]
- /_search/point_in_time
- /_search profile
- /_shard_stores
- /_shrink[5]
- /_snapshot
- /_split
- /_stats
- /_status
- /_tasks
- /_template
- /_update_by_query [1]
- /_validate

- indices.breaker.total.limit
- cluster.max_shards_per_node
- /_cluster/state
- /_cluster/stats
- /_count
- /_dashboards
- /_plugins/_security_analytics
- /_plugins/_sm
- /_plugins/_sql
- /_percolate
- /_rank_eval

1. Perubahan konfigurasi klaster mungkin mengganggu operasi ini sebelum selesai. Kami menyarankan agar Anda menggunakan operasi `/_tasks` bersama dengan operasi ini untuk memverifikasi bahwa permintaan berhasil diselesaikan.
2. HAPUS permintaan untuk `/_search/scroll` dengan badan pesan harus menentukan `"Content-Length"` di header HTTP. Kebanyakan klien menambahkan header ini secara default. Untuk menghindari masalah dengan = karakter dalam `scroll_id` nilai, gunakan badan permintaan, bukan string kueri, untuk meneruskan `scroll_id` nilai ke OpenSearch Service.
3. Untuk pertimbangan tentang menggunakan penulisan, lihat the section called “Sumber daya lain yang didukung”.
4. Lihat metode PUT. Untuk informasi selengkapnya tentang metode GET, lihat the section called “Perbedaan API yang mencolok”. Daftar ini hanya mengacu pada OpenSearch operasi generik yang didukung OpenSearch Layanan dan tidak menyertakan operasi yang didukung khusus plugin untuk deteksi anomali, ISM, dan sebagainya.
5. Lihat the section called “Kecilkan”.

OpenSearch versi 2.3

Untuk OpenSearch 2.3, OpenSearch Layanan mendukung operasi berikut. Untuk informasi tentang sebagian besar operasi, lihat referensi OpenSearch REST API, atau referensi API untuk plugin tertentu.

- Semua operasi di jalur indeks (seperti */index-name* / _forcemerge , */index-nam*
- /_delete_by_query [1]
- /_explain
- /_field_caps
- /_refresh
- /_reindex [1]
- /_render

e /update/*id*, dan /*index-name* /_close)
- /_alias
- /_aliases
- /_all
- /_analyze
- /_bulk
- /_cat (kecuali /_cat/nodeattrs)
- /_cluster/allocation/explain
- /_cluster/health
- /_cluster/pending_tasks
- /_cluster/settings untuk beberapa properti[4]:
  - action.auto_create_index
  - action.search.shard_count.limit
  - indices.breaker.fielddata.limit
  - indices.breaker.request.limit
  - indices.breaker.total.limit
  - cluster.max_shards_per_node
- /_cluster/state
- /_cluster/stats
- /_count
- /_dashboards
- /_field_stats
- /_flush
- /_ingest/pipeline
- /_ltr
- /_mapping
- /_mget
- /_msearch
- /_mtermvectors
- /_nodes
- /_plugins/_asynchronous_search
- /_plugins/_alerting
- /_plugins/_anomaly_detection
- /_plugins/_ism
- /_plugins/_ml
- _plugins/_notifications
- /_plugins/_ppl
- /_plugins/_security
- /_plugins/_sql
- /_percolate
- /_rank_eval
- /_resolve/index
- /_rollover
- /_scripts [3]
- /_search[2]
- /_search profile
- /_shard_stores
- /_shrink[5]
- /_snapshot
- /_split
- /_stats
- /_status
- /_tasks
- /_template
- /_update_by_query [1]
- /_validate

1. Perubahan konfigurasi klaster mungkin mengganggu operasi ini sebelum selesai. Kami menyarankan agar Anda menggunakan operasi `/_tasks` bersama dengan operasi ini untuk memverifikasi bahwa permintaan berhasil diselesaikan.
2. HAPUS permintaan untuk `/_search/scroll` dengan badan pesan harus menentukan `"Content-Length"` di header HTTP. Kebanyakan klien menambahkan header ini secara default. Untuk menghindari masalah dengan = karakter dalam `scroll_id` nilai, gunakan badan permintaan, bukan string kueri, untuk meneruskan `scroll_id` nilai ke OpenSearch Service.
3. Untuk pertimbangan tentang menggunakan penulisan, lihat the section called “Sumber daya lain yang didukung”.
4. Lihat metode PUT. Untuk informasi selengkapnya tentang metode GET, lihat the section called “Perbedaan API yang mencolok”. Daftar ini hanya mengacu pada OpenSearch operasi generik yang didukung OpenSearch Layanan dan tidak menyertakan operasi yang didukung khusus plugin untuk deteksi anomali, ISM, dan sebagainya.
5. Lihat the section called “Kecilkan”.

OpenSearch versi 1.3

Untuk OpenSearch 1.3, OpenSearch Layanan mendukung operasi berikut. Untuk informasi tentang sebagian besar operasi, lihat referensi OpenSearch REST API, atau referensi API untuk plugin tertentu.

- Semua operasi di jalur indeks (seperti */index-name* `/_forcemerge` , */index-name* `/update/`*id*, dan */index-name* `/_close`)
- `/_alias`
- `/_aliases`
- `/_all`
- `/_analyze`
- `/_bulk`
- `/_cat` (kecuali `/_cat/nodeattrs`)
- `/_delete_by_query` [1]
- `/_explain`
- `/_field_caps`
- `/_field_stats`
- `/_flush`
- `/_ingest/pipeline`
- `/_ltr`
- `/_mapping`
- `/_mget`
- `/_msearch`
- `/_mtermvectors`
- `/_nodes`
- `/_refresh`
- `/_reindex` [1]
- `/_render`
- `/_resolve/index`
- `/_rollover`
- `/_scripts` [3]
- `/_search`[2]
- `/_search profile`
- `/_shard_stores`
- `/_shrink`[5]
- `/_snapshot`
- `/_split`

- `/_cluster/allocation/explain`
- `/_cluster/health`
- `/_cluster/pending_tasks`
- `/_cluster/settings` untuk beberapa properti[4]:
  - `action.auto_create_index`
  - `action.search.shard_count.limit`
  - `indices.breaker.fielddata.limit`
  - `indices.breaker.request.limit`
  - `indices.breaker.total.limit`
  - `cluster.max_shards_per_node`
- `/_cluster/state`
- `/_cluster/stats`
- `/_count`
- `/_dashboards`
- `/_plugins/_asynchronous_search`
- `/_plugins/_alerting`
- `/_plugins/_anomaly_detection`
- `/_plugins/_ism`
- `/_plugins/_ml`
- `/_plugins/_ppl`
- `/_plugins/_security`
- `/_plugins/_sql`
- `/_percolate`
- `/_rank_eval`
- `/_stats`
- `/_status`
- `/_tasks`
- `/_template`
- `/_update_by_query` [1]
- `/_validate`

1. Perubahan konfigurasi klaster mungkin mengganggu operasi ini sebelum selesai. Kami menyarankan agar Anda menggunakan operasi `/_tasks` bersama dengan operasi ini untuk memverifikasi bahwa permintaan berhasil diselesaikan.
2. HAPUS permintaan untuk `/_search/scroll` dengan badan pesan harus menentukan `"Content-Length"` di header HTTP. Kebanyakan klien menambahkan header ini secara default. Untuk menghindari masalah dengan = karakter dalam `scroll_id` nilai, gunakan badan permintaan, bukan string kueri, untuk meneruskan `scroll_id` nilai ke OpenSearch Service.
3. Untuk pertimbangan tentang menggunakan penulisan, lihat the section called “Sumber daya lain yang didukung”.

4. Lihat metode PUT. Untuk informasi selengkapnya tentang metode GET, lihat the section called "Perbedaan API yang mencolok". Daftar ini hanya mengacu pada OpenSearch operasi generik yang didukung OpenSearch Layanan dan tidak menyertakan operasi yang didukung khusus plugin untuk deteksi anomali, ISM, dan sebagainya.
5. Lihat the section called "Kecilkan".

OpenSearch versi 1.2

Untuk OpenSearch 1.2, OpenSearch Layanan mendukung operasi berikut. Untuk informasi tentang sebagian besar operasi, lihat referensi OpenSearch REST API, atau referensi API untuk plugin tertentu.

- Semua operasi di jalur indeks (seperti */index-name* /_forcemerge , */index-name* /update/*id*, dan */index-name* /_close)
- /_alias
- /_aliases
- /_all
- /_analyze
- /_bulk
- /_cat (kecuali /_cat/nodeattrs)
- /_cluster/allocation/explain
- /_cluster/health
- /_cluster/pending_tasks
- /_cluster/settings untuk beberapa properti[4]:
  - action.auto_create_index
- /_delete_by_query [1]
- /_explain
- /_field_caps
- /_field_stats
- /_flush
- /_ingest/pipeline
- /_ltr
- /_mapping
- /_mget
- /_msearch
- /_mtermvectors
- /_nodes
- /_plugins/_asynchronous_search
- /_plugins/_alerting
- /_plugins/_anomaly_detection
- /_plugins/_ism
- /_plugins/_ppl
- /_refresh
- /_reindex [1]
- /_render
- /_resolve/index
- /_rollover
- /_scripts [3]
- /_search[2]
- /_search profile
- /_shard_stores
- /_shrink[5]
- /_snapshot
- /_split
- /_stats
- /_status
- /_tasks
- /_template
- /_update_by_query [1]
- /_validate

  - action.search.shard_count.limit
  - indices.breaker.fielddata.limit
  - indices.breaker.request.limit
  - indices.breaker.total.limit
  - cluster.max_shards_per_node
- /_cluster/state
- /_cluster/stats
- /_count
- /_dashboards
- /_plugins/_security
- /_plugins/_sql
- /_percolate
- /_rank_eval

1. Perubahan konfigurasi klaster mungkin mengganggu operasi ini sebelum selesai. Kami menyarankan agar Anda menggunakan operasi `/_tasks` bersama dengan operasi ini untuk memverifikasi bahwa permintaan berhasil diselesaikan.
2. HAPUS permintaan untuk `/_search/scroll` dengan badan pesan harus menentukan `"Content-Length"` di header HTTP. Kebanyakan klien menambahkan header ini secara default. Untuk menghindari masalah dengan = karakter dalam `scroll_id` nilai, gunakan badan permintaan, bukan string kueri, untuk meneruskan `scroll_id` nilai ke OpenSearch Service.
3. Untuk pertimbangan tentang menggunakan penulisan, lihat the section called “Sumber daya lain yang didukung”.
4. Lihat metode `PUT`. Untuk informasi selengkapnya tentang metode `GET`, lihat the section called “Perbedaan API yang mencolok”. Daftar ini hanya mengacu pada OpenSearch operasi generik yang didukung OpenSearch Layanan dan tidak menyertakan operasi yang didukung khusus plugin untuk deteksi anomali, ISM, dan sebagainya.
5. Lihat the section called “Kecilkan”.

OpenSearch versi 1.1

Untuk OpenSearch 1.1, OpenSearch Layanan mendukung operasi berikut. Untuk informasi tentang sebagian besar operasi, lihat referensi OpenSearch REST API, atau referensi API untuk plugin tertentu.

- Semua operasi di jalur indeks (seperti `/index-name /_forcemerge` , `/index-name /update/id`, dan `/index-name /_close`)
- `/_alias`
- `/_aliases`
- `/_all`
- `/_analyze`
- `/_bulk`
- `/_cat` (kecuali `/_cat/nodeattrs`)
- `/_cluster/allocation/explain`
- `/_cluster/health`
- `/_cluster/pending_tasks`
- `/_cluster/settings` untuk beberapa properti[4]:
  - `action.auto_create_index`
  - `action.search.shard_count.limit`
  - `indices.breaker.fielddata.limit`
  - `indices.breaker.request.limit`
- `/_delete_by_query` [1]
- `/_explain`
- `/_field_caps`
- `/_field_stats`
- `/_flush`
- `/_ingest/pipeline`
- `/_ltr`
- `/_mapping`
- `/_mget`
- `/_msearch`
- `/_mtermvectors`
- `/_nodes`
- `/_plugins/_asynchronous_search`
- `/_plugins/_alerting`
- `/_plugins/_anomaly_detection`
- `/_plugins/_ism`
- `/_plugins/_ppl`
- `/_plugins/_security`
- `/_plugins/_sql`
- `/_plugins/_transforms`
- `/_percolate`
- `/_refresh`
- `/_reindex` [1]
- `/_render`
- `/_resolve/index`
- `/_rollover`
- `/_scripts` [3]
- `/_search`[2]
- `/_search profile`
- `/_shard_stores`
- `/_shrink`[5]
- `/_snapshot`
- `/_split`
- `/_stats`
- `/_status`
- `/_tasks`
- `/_template`
- `/_update_by_query` [1]
- `/_validate`

- indices.breaker.total.limit
- cluster.max_shards_per_node
- /_cluster/state
- /_cluster/stats
- /_count
- /_dashboards

- /_rank_eval

1. Perubahan konfigurasi klaster mungkin mengganggu operasi ini sebelum selesai. Kami menyarankan agar Anda menggunakan operasi `/_tasks` bersama dengan operasi ini untuk memverifikasi bahwa permintaan berhasil diselesaikan.
2. HAPUS permintaan untuk `/_search/scroll` dengan badan pesan harus menentukan `"Content-Length"` di header HTTP. Kebanyakan klien menambahkan header ini secara default. Untuk menghindari masalah dengan = karakter dalam `scroll_id` nilai, gunakan badan permintaan, bukan string kueri, untuk meneruskan `scroll_id` nilai ke OpenSearch Service.
3. Untuk pertimbangan tentang menggunakan penulisan, lihat the section called “Sumber daya lain yang didukung”.
4. Lihat metode PUT. Untuk informasi selengkapnya tentang metode GET, lihat the section called “Perbedaan API yang mencolok”. Daftar ini hanya mengacu pada OpenSearch operasi generik yang didukung OpenSearch Layanan dan tidak menyertakan operasi yang didukung khusus plugin untuk deteksi anomali, ISM, dan sebagainya.
5. Lihat the section called “Kecilkan”.

OpenSearch versi 1.0

Untuk OpenSearch 1.0, OpenSearch Layanan mendukung operasi berikut. Untuk informasi tentang sebagian besar operasi, lihat referensi OpenSearch REST API, atau referensi API untuk plugin tertentu.

- Semua operasi di jalur indeks (seperti */index-name* / _forcemerge , */index-nam*
- /_delete_by_query [1]
- /_explain
- /_field_caps
- /_refresh
- /_reindex [1]
- /_render

e /update/*id*, dan /*index-name* /_close)
- /_alias
- /_aliases
- /_all
- /_analyze
- /_bulk
- /_cat (kecuali /_cat/nodeattrs)
- /_cluster/allocation/explain
- /_cluster/health
- /_cluster/pending_tasks
- /_cluster/settings untuk beberapa properti[4]:
  - action.auto_create_index
  - action.search.shard_count.limit
  - indices.breaker.fielddata.limit
  - indices.breaker.request.limit
  - indices.breaker.total.limit
  - cluster.max_shards_per_node
- /_cluster/state
- /_cluster/stats
- /_count
- /_dashboards
- /_field_stats
- /_flush
- /_ingest/pipeline
- /_ltr
- /_mapping
- /_mget
- /_msearch
- /_mtermvectors
- /_nodes
- /_plugins/_asynchronous_search
- /_plugins/_alerting
- /_plugins/_anomaly_detection
- /_plugins/_ism
- /_plugins/_ppl
- /_plugins/_security
- /_plugins/_sql
- /_plugins/_transforms
- /_percolate
- /_rank_eval
- /_resolve/index
- /_rollover
- /_scripts [3]
- /_search[2]
- /_search profile
- /_shard_stores
- /_shrink[5]
- /_snapshot
- /_split
- /_stats
- /_status
- /_tasks
- /_template
- /_update_by_query [1]
- /_validate

1. Perubahan konfigurasi klaster mungkin mengganggu operasi ini sebelum selesai. Kami menyarankan agar Anda menggunakan operasi `/_tasks` bersama dengan operasi ini untuk memverifikasi bahwa permintaan berhasil diselesaikan.
2. HAPUS permintaan untuk `/_search/scroll` dengan badan pesan harus menentukan `"Content-Length"` di header HTTP. Kebanyakan klien menambahkan header ini secara default. Untuk menghindari masalah dengan = karakter dalam `scroll_id` nilai, gunakan badan permintaan, bukan string kueri, untuk meneruskan `scroll_id` nilai ke OpenSearch Service.
3. Untuk pertimbangan tentang menggunakan penulisan, lihat the section called “Sumber daya lain yang didukung”.
4. Lihat metode `PUT`. Untuk informasi selengkapnya tentang metode `GET`, lihat the section called “Perbedaan API yang mencolok”. Daftar ini hanya mengacu pada OpenSearch operasi generik yang didukung OpenSearch Layanan dan tidak menyertakan operasi yang didukung khusus plugin untuk deteksi anomali, ISM, dan sebagainya.
5. Lihat the section called “Kecilkan”.

Elasticsearch versi 7.10

Untuk Elasticsearch 7.10, OpenSearch Layanan mendukung operasi berikut.

- Semua operasi di jalur indeks (seperti */index-name* `/_forcemerge` , */index-name* `/update/`*id*, dan */index-name* `/_close`)
- `/_alias`
- `/_aliases`
- `/_all`
- `/_analyze`
- `/_bulk`
- `/_cat` (kecuali `/_cat/nodeattrs`)
- `/_cluster/allocation/explain`
- `/_cluster/health`
- `/_delete_by_query` [1]
- `/_explain`
- `/_field_caps`
- `/_field_stats`
- `/_flush`
- `/_index_template` [6]
- `/_ingest/pipeline`
- `/_index_template`
- `/_ltr`
- `/_mapping`
- `/_mget`
- `/_msearch`
- `/_mtermvectors`
- `/_nodes`
- `/_refresh`
- `/_reindex` [1]
- `/_render`
- `/_resolve/index`
- `/_rollover`
- `/_scripts` [3]
- `/_search`[2]
- `/_search profile`
- `/_shard_stores`
- `/_shrink`[5]
- `/_snapshot`
- `/_split`
- `/_stats`
- `/_status`

- `/_cluster/pending_tasks`
- `/_cluster/settings` untuk beberapa properti[4]:
  - `action.auto_create_index`
  - `action.search.shard_count.limit`
  - `indices.breaker.fielddata.limit`
  - `indices.breaker.request.limit`
  - `indices.breaker.total.limit`
  - `cluster.max_shards_per_node`
- `/_cluster/state`
- `/_cluster/stats`
- `/_count`
- `/_opendistro/_alerting`
- `/_opendistro/_asynchronous_search`
- `/_opendistro/_anomaly_detection`
- `/_opendistro/_ism`
- `/_opendistro/_ppl`
- `/_opendistro/_security`
- `/_opendistro/_sql`
- `/_percolate`
- `/_plugin/kibana`
- `/_plugins/_replication`
- `/_rank_eval`
- `/_tasks`
- `/_template` [6]
- `/_update_by_query` [1]
- `/_validate`

1. Perubahan konfigurasi klaster mungkin mengganggu operasi ini sebelum selesai. Kami menyarankan agar Anda menggunakan operasi `/_tasks` bersama dengan operasi ini untuk memverifikasi bahwa permintaan berhasil diselesaikan.
2. HAPUS permintaan untuk `/_search/scroll` dengan badan pesan harus menentukan `"Content-Length"` di header HTTP. Kebanyakan klien menambahkan header ini secara default. Untuk menghindari masalah dengan = karakter dalam `scroll_id` nilai, gunakan badan permintaan, bukan string kueri, untuk meneruskan `scroll_id` nilai ke OpenSearch Service.
3. Untuk pertimbangan tentang menggunakan penulisan, lihat the section called “Sumber daya lain yang didukung”.
4. Lihat metode `PUT`. Untuk informasi selengkapnya tentang metode `GET`, lihat the section called “Perbedaan API yang mencolok”. Daftar ini hanya mengacu pada operasi Elasticsearch generik yang didukung OpenSearch Layanan dan tidak menyertakan operasi yang didukung khusus plugin untuk deteksi anomali, ISM, dan sebagainya.
5. Lihat the section called “Kecilkan”.

6. Template indeks lama (`_template`) digantikan oleh composable templates (`_index_template`) dimulai dengan Elasticsearch 7.8. Template yang dapat dikomposisi lebih diutamakan daripada templat lama. Jika tidak ada template composable yang cocok dengan indeks tertentu, template lama masih dapat cocok dan diterapkan. `_template`Operasi masih berfungsi pada OpenSearch dan versi Elasticsearch OSS yang lebih baru, tetapi panggilan GET ke dua jenis template mengembalikan hasil yang berbeda.

Elasticsearch versi 7.9

Untuk Elasticsearch 7.9, OpenSearch Layanan mendukung operasi berikut.

- Semua operasi di jalur indeks (seperti */index-name* / `_forcemerge` , */index-name* /`update/`*id*, dan */index-name* /`_close`)
- `/_alias`
- `/_aliases`
- `/_all`
- `/_analyze`
- `/_bulk`
- `/_cat` (kecuali `/_cat/nodeattrs`)
- `/_cluster/allocation/explain`
- `/_cluster/health`
- `/_cluster/pending_tasks`
- `/_cluster/settings` untuk beberapa properti[4]:
  - `action.auto_create_index`
  - `action.search.shard_count.limit`
- `/_delete_by_query` [1]
- `/_explain`
- `/_field_caps`
- `/_field_stats`
- `/_flush`
- `/_index_template` [6]
- `/_ingest/pipeline`
- `/_ltr`
- `/_mapping`
- `/_mget`
- `/_msearch`
- `/_mtermvectors`
- `/_nodes`
- `/_opendistro/_alerting`
- `/_opendistro/_anomaly_detection`
- `/_opendistro/_ism`
- `/_opendistro/_ppl`
- `/_opendistro/_security`
- `/_refresh`
- `/_reindex` [1]
- `/_render`
- `/_resolve/index`
- `/_rollover`
- `/_scripts` [3]
- `/_search`[2]
- `/_search profile`
- `/_shard_stores`
- `/_shrink`[5]
- `/_snapshot`
- `/_split`
- `/_stats`
- `/_status`
- `/_tasks`
- `/_template` [6]
- `/_update_by_query` [1]
- `/_validate`

  - indices.breaker.fielddata.limit
  - indices.breaker.request.limit
  - indices.breaker.total.limit
  - cluster.max_shards_per_node
- /_cluster/state
- /_cluster/stats
- /_count
- /_opendistro/_sql
- /_percolate
- /_plugin/kibana
- /_rank_eval

1. Perubahan konfigurasi klaster mungkin mengganggu operasi ini sebelum selesai. Kami menyarankan agar Anda menggunakan operasi `/_tasks` bersama dengan operasi ini untuk memverifikasi bahwa permintaan berhasil diselesaikan.
2. HAPUS permintaan untuk `/_search/scroll` dengan badan pesan harus menentukan `"Content-Length"` di header HTTP. Kebanyakan klien menambahkan header ini secara default. Untuk menghindari masalah dengan = karakter dalam `scroll_id` nilai, gunakan badan permintaan, bukan string kueri, untuk meneruskan `scroll_id` nilai ke OpenSearch Service.
3. Untuk pertimbangan tentang menggunakan penulisan, lihat the section called “Sumber daya lain yang didukung”.
4. Lihat metode PUT. Untuk informasi selengkapnya tentang metode GET, lihat the section called “Perbedaan API yang mencolok”. Daftar ini hanya mengacu pada OpenSearch operasi generik yang didukung OpenSearch Layanan dan tidak menyertakan operasi yang didukung khusus plugin untuk deteksi anomali, ISM, dan sebagainya.
5. Lihat the section called “Kecilkan”.
6. Template indeks lama (`_template`) digantikan oleh composable templates (`_index_template`) dimulai dengan Elasticsearch 7.8. Template yang dapat dikomposisi lebih diutamakan daripada templat lama. Jika tidak ada template composable yang cocok dengan indeks tertentu, template lama masih dapat cocok dan diterapkan. `_template`Operasi masih berfungsi pada OpenSearch dan versi Elasticsearch OSS yang lebih baru, tetapi panggilan GET ke dua jenis template mengembalikan hasil yang berbeda.

Elasticsearch versi 7.8

Untuk Elasticsearch 7.8, OpenSearch Layanan mendukung operasi berikut.

- Semua operasi di jalur indeks (seperti */index-name* / _forcemerge , */index-name* /update/*id*, dan */index-name* /_close)
- /_alias
- /_aliases
- /_all
- /_analyze
- /_bulk
- /_cat (kecuali /_cat/nodeattrs)
- /_cluster/allocation/explain
- /_cluster/health
- /_cluster/pending_tasks
- /_cluster/settings untuk beberapa properti[4]:
  - action.auto_create_index
  - action.search.shard_count.limit
  - indices.breaker.fielddata.limit
  - indices.breaker.request.limit
  - indices.breaker.total.limit
- /_cluster/state
- /_cluster/stats
- /_count
- /_delete_by_query [1]
- /_explain
- /_field_caps
- /_field_stats
- /_flush
- /_index_template [6]
- /_ingest/pipeline
- /_ltr
- /_mapping
- /_mget
- /_msearch
- /_mtermvectors
- /_nodes
- /_opendistro/_alerting
- /_opendistro/_anomaly_detection
- /_opendistro/_ism
- /_opendistro/_security
- /_opendistro/_sql
- /_percolate
- /_plugin/kibana
- /_rank_eval
- /_refresh
- /_reindex [1]
- /_render
- /_rollover
- /_scripts [3]
- /_search[2]
- /_search profile
- /_shard_stores
- /_shrink[5]
- /_snapshot
- /_split
- /_stats
- /_status
- /_tasks
- /_template [6]
- /_update_by_query [1]
- /_validate

- `cluster.max_shards_per_node`

1. Perubahan konfigurasi klaster mungkin mengganggu operasi ini sebelum selesai. Kami menyarankan agar Anda menggunakan operasi `/_tasks` bersama dengan operasi ini untuk memverifikasi bahwa permintaan berhasil diselesaikan.
2. HAPUS permintaan untuk `/_search/scroll` dengan badan pesan harus menentukan `"Content-Length"` di header HTTP. Kebanyakan klien menambahkan header ini secara default. Untuk menghindari masalah dengan = karakter dalam `scroll_id` nilai, gunakan badan permintaan, bukan string kueri, untuk meneruskan `scroll_id` nilai ke OpenSearch Service.
3. Untuk pertimbangan tentang menggunakan penulisan, lihat the section called “Sumber daya lain yang didukung”.
4. Lihat metode PUT. Untuk informasi selengkapnya tentang metode GET, lihat the section called “Perbedaan API yang mencolok”. Daftar ini hanya mengacu pada operasi Elasticsearch generik yang didukung OpenSearch Layanan dan tidak menyertakan operasi yang didukung khusus plugin untuk deteksi anomali, ISM, dan sebagainya.
5. Lihat the section called “Kecilkan”.
6. Template indeks lama (`_template`) digantikan oleh composable templates (`_index_template`) dimulai dengan Elasticsearch 7.8. Template yang dapat dikomposisi lebih diutamakan daripada templat lama. Jika tidak ada template composable yang cocok dengan indeks tertentu, template lama masih dapat cocok dan diterapkan. `_template`Operasi masih berfungsi pada OpenSearch dan versi Elasticsearch OSS yang lebih baru, tetapi panggilan GET ke dua jenis template mengembalikan hasil yang berbeda.

Elasticsearch versi 7.7

Untuk Elasticsearch 7.7, OpenSearch Layanan mendukung operasi berikut.

- Semua operasi di jalur indeks (seperti */index-name* */_forcemerge* , */index-name* */update/id*, dan */index-name* */_close*)
- `/_alias`
- `/_cluster/state`
- `/_cluster/stats`
- `/_count`
- `/_delete_by_query` [1]
- `/_explain`
- `/_refresh`
- `/_reindex` [1]
- `/_render`
- `/_rollover`
- `/_scripts` [3]

- /_aliases
- /_all
- /_analyze
- /_bulk
- /_cat (kecuali /_cat/nodeattrs)
- /_cluster/allocation/explain
- /_cluster/health
- /_cluster/pending_tasks
- /_cluster/settings untuk beberapa properti[4]:
  - action.auto_create_index
  - action.search.shard_count.limit
  - indices.breaker.fielddata.limit
  - indices.breaker.request.limit
  - indices.breaker.total.limit
  - cluster.max_shards_per_node
- /_field_caps
- /_field_stats
- /_flush
- /_ingest/pipeline
- /_ltr
- /_mapping
- /_mget
- /_msearch
- /_mtermvectors
- /_nodes
- /_opendistro/_alerting
- /_opendistro/_anomaly_detection
- /_opendistro/_ism
- /_opendistro/_security
- /_opendistro/_sql
- /_percolate
- /_plugin/kibana
- /_rank_eval
- /_search[2]
- /_search profile
- /_shard_stores
- /_shrink[5]
- /_snapshot
- /_split
- /_stats
- /_status
- /_tasks
- /_template
- /_update_by_query [1]
- /_validate

1. Perubahan konfigurasi klaster mungkin mengganggu operasi ini sebelum selesai. Kami menyarankan agar Anda menggunakan operasi /_tasks bersama dengan operasi ini untuk memverifikasi bahwa permintaan berhasil diselesaikan.
2. HAPUS permintaan untuk /_search/scroll dengan badan pesan harus menentukan "Content-Length" di header HTTP. Kebanyakan klien menambahkan header ini secara default. Untuk menghindari masalah dengan = karakter dalam scroll_id nilai, gunakan badan permintaan, bukan string kueri, untuk meneruskan scroll_id nilai ke OpenSearch Service.

3. Untuk pertimbangan tentang menggunakan penulisan, lihat the section called “Sumber daya lain yang didukung”.
4. Lihat metode PUT. Untuk informasi selengkapnya tentang metode GET, lihat the section called “Perbedaan API yang mencolok”. Daftar ini hanya mengacu pada operasi Elasticsearch generik yang didukung OpenSearch Layanan dan tidak menyertakan operasi yang didukung khusus plugin untuk deteksi anomali, ISM, dan sebagainya.
5. Lihat the section called “Kecilkan”.

Elasticsearch versi 7.4

Untuk Elasticsearch 7.4, OpenSearch Layanan mendukung operasi berikut.

- Semua operasi di jalur indeks (seperti */index-name* / _forcemerge , */index-name* /update/*id*, dan */index-name* /_close)
- /_alias
- /_aliases
- /_all
- /_analyze
- /_bulk
- /_cat (kecuali /_cat/nodeattrs)
- /_cluster/allocation/explain
- /_cluster/health
- /_cluster/pending_tasks
- /_cluster/settings untuk beberapa properti[4]:
  - action.auto_create_index
- /_cluster/state
- /_cluster/stats
- /_count
- /_delete_by_query [1]
- /_explain
- /_field_caps
- /_field_stats
- /_flush
- /_ingest/pipeline
- /_mapping
- /_mget
- /_msearch
- /_mtermvectors
- /_nodes
- /_opendistro/_alerting
- /_opendistro/_anomaly_detection
- /_opendistro/_ism
- /_refresh
- /_reindex [1]
- /_render
- /_rollover
- /_scripts [3]
- /_search[2]
- /_search profile
- /_shard_stores
- /_shrink[5]
- /_snapshot
- /_split
- /_stats
- /_status
- /_tasks
- /_template
- /_update_by_query [1]
- /_validate

- `action.search.shard_count.limit`
- `indices.breaker.fielddata.limit`
- `indices.breaker.request.limit`
- `indices.breaker.total.limit`
- `cluster.max_shards_per_node`
- `/_opendistro/_security`
- `/_opendistro/_sql`
- `/_percolate`
- `/_plugin/kibana`
- `/_rank_eval`

1. Perubahan konfigurasi klaster mungkin mengganggu operasi ini sebelum selesai. Kami menyarankan agar Anda menggunakan operasi `/_tasks` bersama dengan operasi ini untuk memverifikasi bahwa permintaan berhasil diselesaikan.
2. HAPUS permintaan untuk `/_search/scroll` dengan badan pesan harus menentukan `"Content-Length"` di header HTTP. Kebanyakan klien menambahkan header ini secara default. Untuk menghindari masalah dengan = karakter dalam `scroll_id` nilai, gunakan badan permintaan, bukan string kueri, untuk meneruskan `scroll_id` nilai ke OpenSearch Service.
3. Untuk pertimbangan tentang menggunakan penulisan, lihat the section called “Sumber daya lain yang didukung”.
4. Lihat metode PUT. Untuk informasi selengkapnya tentang metode GET, lihat the section called “Perbedaan API yang mencolok”. Daftar ini hanya mengacu pada operasi Elasticsearch generik yang didukung OpenSearch Layanan dan tidak menyertakan operasi yang didukung khusus plugin untuk deteksi anomali, ISM, dan sebagainya.
5. Lihat the section called “Kecilkan”.

Elasticsearch versi 7.1

Untuk Elasticsearch 7.1, OpenSearch Layanan mendukung operasi berikut.

- Semua operasi di jalur indeks (seperti */index-name* `/_forcemerge` dan */index-*
- `/_cluster/state`
- `/_cluster/stats`
- `/_count`
- `/_refresh`
- `/_reindex` [1]
- `/_render`

name /update/*id*) kecuali /*index-name* /_close
- /_alias
- /_aliases
- /_all
- /_analyze
- /_bulk
- /_cat (kecuali /_cat/nodeattrs)
- /_cluster/allocation/explain
- /_cluster/health
- /_cluster/pending_tasks
- /_cluster/settings untuk beberapa properti[4]:
  - action.auto_create_index
  - action.search.shard_count.limit
  - indices.breaker.fielddata.limit
  - indices.breaker.request.limit
  - indices.breaker.total.limit
  - cluster.max_shards_per_node
- /_delete_by_query [1]
- /_explain
- /_field_caps
- /_field_stats
- /_flush
- /_ingest/pipeline
- /_mapping
- /_mget
- /_msearch
- /_mtermvectors
- /_nodes
- /_opendistro/_alerting
- /_opendistro/_ism
- /_opendistro/_security
- /_opendistro/_sql
- /_percolate
- /_plugin/kibana
- /_rank_eval
- /_rollover
- /_scripts [3]
- /_search[2]
- /_search profile
- /_shard_stores
- /_shrink[5]
- /_snapshot
- /_split
- /_stats
- /_status
- /_tasks
- /_template
- /_update_by_query [1]
- /_validate

1. Perubahan konfigurasi klaster mungkin mengganggu operasi ini sebelum selesai. Kami menyarankan agar Anda menggunakan operasi /_tasks bersama dengan operasi ini untuk memverifikasi bahwa permintaan berhasil diselesaikan.

2. HAPUS permintaan untuk `/_search/scroll` dengan badan pesan harus menentukan `"Content-Length"` di header HTTP. Kebanyakan klien menambahkan header ini secara default. Untuk menghindari masalah dengan = karakter dalam `scroll_id` nilai, gunakan badan permintaan, bukan string kueri, untuk meneruskan `scroll_id` nilai ke OpenSearch Service.
3. Untuk pertimbangan tentang menggunakan penulisan, lihat the section called “Sumber daya lain yang didukung”.
4. Lihat metode PUT. Untuk informasi selengkapnya tentang metode GET, lihat the section called “Perbedaan API yang mencolok”. Daftar ini hanya mengacu pada operasi Elasticsearch generik yang didukung OpenSearch Layanan dan tidak menyertakan operasi yang didukung khusus plugin untuk deteksi anomali, ISM, dan sebagainya.
5. Lihat the section called “Kecilkan”.

Elasticsearch versi 6.8

Untuk Elasticsearch 6.8, OpenSearch Layanan mendukung operasi berikut.

- Semua operasi di jalur indeks (seperti */index-name* */_forcemerge* dan */index-name* */update/id*) kecuali */index-name* */_close*
- `/_alias`
- `/_aliases`
- `/_all`
- `/_analyze`
- `/_bulk`
- `/_cat` (kecuali `/_cat/nodeattrs`)
- `/_cluster/allocation/explain`
- `/_cluster/health`
- `/_cluster/pending_tasks`
- `/_cluster/state`
- `/_cluster/stats`
- `/_count`
- `/_delete_by_query` [1]
- `/_explain`
- `/_field_caps`
- `/_field_stats`
- `/_flush`
- `/_ingest/pipeline`
- `/_mapping`
- `/_mget`
- `/_msearch`
- `/_mtermvectors`
- `/_nodes`
- `/_opendistro/_alerting`
- `/_refresh`
- `/_reindex` [1]
- `/_render`
- `/_rollover`
- `/_scripts` [3]
- `/_search`[2]
- `/_search profile`
- `/_shard_stores`
- `/_shrink`[5]
- `/_snapshot`
- `/_split`
- `/_stats`
- `/_status`
- `/_tasks`
- `/_template`
- `/_update_by_query` [1]

- `/_cluster/settings` untuk beberapa properti[4]:
  - `action.auto_create_index`
  - `action.search.shard_count.limit`
  - `indices.breaker.fielddata.limit`
  - `indices.breaker.request.limit`
  - `indices.breaker.total.limit`
  - `cluster.max_shards_per_node`
  - `cluster.blocks.read_only`
- `/_opendistro/_ism`
- `/_opendistro/_security`
- `/_opendistro/_sql`
- `/_percolate`
- `/_plugin/kibana`
- `/_rank_eval`
- `/_validate`

1. Perubahan konfigurasi klaster mungkin mengganggu operasi ini sebelum selesai. Kami menyarankan agar Anda menggunakan operasi `/_tasks` bersama dengan operasi ini untuk memverifikasi bahwa permintaan berhasil diselesaikan.
2. HAPUS permintaan untuk `/_search/scroll` dengan badan pesan harus menentukan `"Content-Length"` di header HTTP. Kebanyakan klien menambahkan header ini secara default. Untuk menghindari masalah dengan = karakter dalam `scroll_id` nilai, gunakan badan permintaan, bukan string kueri, untuk meneruskan `scroll_id` nilai ke OpenSearch Service.
3. Untuk pertimbangan tentang menggunakan penulisan, lihat the section called “Sumber daya lain yang didukung”.
4. Lihat metode PUT. Untuk informasi selengkapnya tentang metode GET, lihat the section called “Perbedaan API yang mencolok”. Daftar ini hanya mengacu pada operasi Elasticsearch generik yang didukung OpenSearch Layanan dan tidak menyertakan operasi yang didukung khusus plugin untuk deteksi anomali, ISM, dan sebagainya.
5. Lihat the section called “Kecilkan”.

Elasticsearch versi 6.7

Untuk Elasticsearch 6.7, OpenSearch Layanan mendukung operasi berikut.

- Semua operasi di jalur indeks (seperti */index-name* / _forcemerge dan */index-name* /update/*id*) kecuali */index-name* /_close
- /_alias
- /_aliases
- /_all
- /_analyze
- /_bulk
- /_cat (kecuali /_cat/nodeattrs)
- /_cluster/allocation/explain
- /_cluster/health
- /_cluster/pending_tasks
- /_cluster/settings untuk beberapa properti[4]:
  - action.auto_create_index
  - action.search.shard_count.limit
  - indices.breaker.fielddata.limit
  - indices.breaker.request.limit
  - indices.breaker.total.limit
- /_cluster/state
- /_cluster/stats
- /_count
- /_delete_by_query [1]
- /_explain
- /_field_caps
- /_field_stats
- /_flush
- /_ingest/pipeline
- /_mapping
- /_mget
- /_msearch
- /_mtermvectors
- /_nodes
- /_opendistro/_alerting
- /_opendistro/_security
- /_opendistro/_sql
- /_percolate
- /_plugin/kibana
- /_rank_eval
- /_refresh
- /_reindex [1]
- /_render
- /_rollover
- /_scripts [3]
- /_search[2]
- /_search profile
- /_shard_stores
- /_shrink[5]
- /_snapshot
- /_split
- /_stats
- /_status
- /_tasks
- /_template
- /_update_by_query [1]
- /_validate

- `cluster.max_shards_per_node`

1. Perubahan konfigurasi klaster mungkin mengganggu operasi ini sebelum selesai. Kami menyarankan agar Anda menggunakan operasi `/_tasks` bersama dengan operasi ini untuk memverifikasi bahwa permintaan berhasil diselesaikan.
2. HAPUS permintaan untuk `/_search/scroll` dengan badan pesan harus menentukan `"Content-Length"` di header HTTP. Kebanyakan klien menambahkan header ini secara default. Untuk menghindari masalah dengan = karakter dalam `scroll_id` nilai, gunakan badan permintaan, bukan string kueri, untuk meneruskan `scroll_id` nilai ke OpenSearch Service.
3. Untuk pertimbangan tentang menggunakan penulisan, lihat the section called “Sumber daya lain yang didukung”.
4. Lihat metode PUT. Untuk informasi selengkapnya tentang metode GET, lihat the section called “Perbedaan API yang mencolok”. Daftar ini hanya mengacu pada operasi Elasticsearch generik yang didukung OpenSearch Layanan dan tidak menyertakan operasi yang didukung khusus plugin untuk deteksi anomali, ISM, dan sebagainya.
5. Lihat the section called “Kecilkan”.

Elasticsearch versi 6.5

Untuk Elasticsearch 6.5, OpenSearch Layanan mendukung operasi berikut.

- Semua operasi di jalur indeks (seperti */index-name* `/_forcemerge` dan */index-name* `/update/`*id*) kecuali */index-name* `/_close`
- `/_alias`
- `/_aliases`
- `/_all`
- `/_analyze`
- `/_bulk`
- `/_cluster/state`
- `/_cluster/stats`
- `/_count`
- `/_delete_by_query` [1]
- `/_explain`
- `/_field_caps`
- `/_field_stats`
- `/_flush`
- `/_ingest/pipeline`
- `/_mapping`
- `/_refresh`
- `/_reindex` [1]
- `/_render`
- `/_rollover`
- `/_scripts` [3]
- `/_search`[2]
- `/_search profile`
- `/_shard_stores`
- `/_shrink`[5]
- `/_snapshot`

- `/_cat` (kecuali `/_cat/nodeattrs`)
- `/_cluster/allocation/explain`
- `/_cluster/health`
- `/_cluster/pending_tasks`
- `/_cluster/settings` untuk beberapa properti[4]:
  - `action.auto_create_index`
  - `action.search.shard_count.limit`
  - `indices.breaker.fielddata.limit`
  - `indices.breaker.request.limit`
  - `indices.breaker.total.limit`
- `/_mget`
- `/_msearch`
- `/_mtermvectors`
- `/_nodes`
- `/_opendistro/_alerting`
- `/_opendistro/_sql`
- `/_percolate`
- `/_plugin/kibana`
- `/_rank_eval`
- `/_split`
- `/_stats`
- `/_status`
- `/_tasks`
- `/_template`
- `/_update_by_query` [1]
- `/_validate`

1. Perubahan konfigurasi klaster mungkin mengganggu operasi ini sebelum selesai. Kami menyarankan agar Anda menggunakan operasi `/_tasks` bersama dengan operasi ini untuk memverifikasi bahwa permintaan berhasil diselesaikan.
2. HAPUS permintaan untuk `/_search/scroll` dengan badan pesan harus menentukan `"Content-Length"` di header HTTP. Kebanyakan klien menambahkan header ini secara default. Untuk menghindari masalah dengan = karakter dalam `scroll_id` nilai, gunakan badan permintaan, bukan string kueri, untuk meneruskan `scroll_id` nilai ke OpenSearch Service.
3. Untuk pertimbangan tentang menggunakan penulisan, lihat the section called “Sumber daya lain yang didukung”.
4. Lihat metode `PUT`. Untuk informasi selengkapnya tentang metode `GET`, lihat the section called “Perbedaan API yang mencolok”. Daftar ini hanya mengacu pada operasi Elasticsearch generik yang didukung OpenSearch Layanan dan tidak menyertakan operasi yang didukung khusus plugin untuk deteksi anomali, ISM, dan sebagainya.
5. Lihat the section called “Kecilkan”.

Elasticsearch versi 6.4

Untuk Elasticsearch 6.4, OpenSearch Layanan mendukung operasi berikut.

- Semua operasi di jalur indeks (seperti */index-name* / _forcemerge dan */index-name* /update/*id*) kecuali */index-name* /_close
- /_alias
- /_aliases
- /_all
- /_analyze
- /_bulk
- /_cat (kecuali /_cat/nodeattrs)
- /_cluster/allocation/explain
- /_cluster/health
- /_cluster/pending_tasks
- /_cluster/settings untuk beberapa properti[4]:
  - action.auto_create_index
  - action.search.shard_count.limit
  - indices.breaker.fielddata.limit
  - indices.breaker.request.limit
  - indices.breaker.total.limit
- /_cluster/state
- /_cluster/stats
- /_count
- /_delete_by_query [1]
- /_explain
- /_field_caps
- /_field_stats
- /_flush
- /_ingest/pipeline
- /_mapping
- /_mget
- /_msearch
- /_mtermvectors
- /_nodes
- /_opendistro/_alerting
- /_percolate
- /_plugin/kibana
- /_rank_eval
- /_refresh
- /_reindex [1]
- /_render
- /_rollover
- /_scripts [3]
- /_search[2]
- /_search profile
- /_shard_stores
- /_shrink[5]
- /_snapshot
- /_split
- /_stats
- /_status
- /_tasks
- /_template
- /_update_by_query [1]
- /_validate

1. Perubahan konfigurasi klaster mungkin mengganggu operasi ini sebelum selesai. Kami menyarankan agar Anda menggunakan operasi `/_tasks` bersama dengan operasi ini untuk memverifikasi bahwa permintaan berhasil diselesaikan.
2. HAPUS permintaan untuk `/_search/scroll` dengan badan pesan harus menentukan `"Content-Length"` di header HTTP. Kebanyakan klien menambahkan header ini secara default. Untuk menghindari masalah dengan = karakter dalam `scroll_id` nilai, gunakan badan permintaan, bukan string kueri, untuk meneruskan `scroll_id` nilai ke OpenSearch Service.
3. Untuk pertimbangan tentang menggunakan penulisan, lihat the section called “Sumber daya lain yang didukung”.
4. Lihat metode `PUT`. Untuk informasi selengkapnya tentang metode `GET`, lihat the section called “Perbedaan API yang mencolok”. Daftar ini hanya mengacu pada operasi Elasticsearch generik yang didukung OpenSearch Layanan dan tidak menyertakan operasi yang didukung khusus plugin untuk deteksi anomali, ISM, dan sebagainya.
5. Lihat the section called “Kecilkan”.

Elasticsearch versi 6.3

Untuk Elasticsearch 6.3, OpenSearch Layanan mendukung operasi berikut.

- Semua operasi di jalur indeks (seperti */index-name /_forcemerge* dan */index-name /update/id*) kecuali */index-name /_close*
- `/_alias`
- `/_aliases`
- `/_all`
- `/_analyze`
- `/_bulk`
- `/_cat` (kecuali `/_cat/nodeattrs`)
- `/_cluster/allocation/explain`
- `/_cluster/health`
- `/_cluster/state`
- `/_cluster/stats`
- `/_count`
- `/_delete_by_query` [1]
- `/_explain`
- `/_field_caps`
- `/_field_stats`
- `/_flush`
- `/_ingest/pipeline`
- `/_mapping`
- `/_mget`
- `/_msearch`
- `/_mtermvectors`
- `/_nodes`
- `/_refresh`
- `/_reindex` [1]
- `/_render`
- `/_rollover`
- `/_scripts` [3]
- `/_search`[2]
- `/_search profile`
- `/_shard_stores`
- `/_shrink`[5]
- `/_snapshot`
- `/_split`
- `/_stats`
- `/_status`
- `/_tasks`

- `/_cluster/pending_tasks`
- `/_cluster/settings` untuk beberapa properti[4]:
  - `action.auto_create_index`
  - `action.search.shard_count.limit`
  - `indices.breaker.fielddata.limit`
  - `indices.breaker.request.limit`
  - `indices.breaker.total.limit`
- `/_opendistro/_alerting`
- `/_percolate`
- `/_plugin/kibana`
- `/_rank_eval`
- `/_template`
- `/_update_by_query` [1]
- `/_validate`

1. Perubahan konfigurasi klaster mungkin mengganggu operasi ini sebelum selesai. Kami menyarankan agar Anda menggunakan operasi `/_tasks` bersama dengan operasi ini untuk memverifikasi bahwa permintaan berhasil diselesaikan.
2. HAPUS permintaan untuk `/_search/scroll` dengan badan pesan harus menentukan `"Content-Length"` di header HTTP. Kebanyakan klien menambahkan header ini secara default. Untuk menghindari masalah dengan = karakter dalam `scroll_id` nilai, gunakan badan permintaan, bukan string kueri, untuk meneruskan `scroll_id` nilai ke OpenSearch Service.
3. Untuk pertimbangan tentang menggunakan penulisan, lihat the section called “Sumber daya lain yang didukung”.
4. Lihat metode `PUT`. Untuk informasi selengkapnya tentang metode `GET`, lihat the section called “Perbedaan API yang mencolok”. Daftar ini hanya mengacu pada operasi Elasticsearch generik yang didukung OpenSearch Layanan dan tidak menyertakan operasi yang didukung khusus plugin untuk deteksi anomali, ISM, dan sebagainya.
5. Lihat the section called “Kecilkan”.

Elasticsearch versi 6.2

Untuk Elasticsearch 6.2, OpenSearch Layanan mendukung operasi berikut.

- Semua operasi di jalur indeks (seperti */index-name* / _forcemerge dan */index-name* /update/*id*) kecuali */index-name* /_close
- /_alias
- /_aliases
- /_all
- /_analyze
- /_bulk
- /_cat (kecuali /_cat/nodeattrs)
- /_cluster/allocation/explain
- /_cluster/health
- /_cluster/pending_tasks
- /_cluster/settings untuk beberapa properti[4]:
  - action.auto_create_index
  - action.search.shard_count.limit
  - indices.breaker.fielddata.limit
  - indices.breaker.request.limit
  - indices.breaker.total.limit
- /_cluster/state
- /_cluster/stats
- /_count
- /_delete_by_query [1]
- /_explain
- /_field_caps
- /_field_stats
- /_flush
- /_ingest/pipeline
- /_mapping
- /_mget
- /_msearch
- /_mtermvectors
- /_nodes
- /_opendistro/_alerting
- /_percolate
- /_plugin/kibana
- /_rank_eval
- /_refresh
- /_reindex [1]
- /_render
- /_rollover
- /_scripts [3]
- /_search[2]
- /_search profile
- /_shard_stores
- /_shrink[5]
- /_snapshot
- /_split
- /_stats
- /_status
- /_tasks
- /_template
- /_update_by_query [1]
- /_validate

1. Perubahan konfigurasi klaster mungkin mengganggu operasi ini sebelum selesai. Kami menyarankan agar Anda menggunakan operasi /_tasks bersama dengan operasi ini untuk memverifikasi bahwa permintaan berhasil diselesaikan.

2. HAPUS permintaan untuk `/_search/scroll` dengan badan pesan harus menentukan `"Content-Length"` di header HTTP. Kebanyakan klien menambahkan header ini secara default. Untuk menghindari masalah dengan = karakter dalam `scroll_id` nilai, gunakan badan permintaan, bukan string kueri, untuk meneruskan `scroll_id` nilai ke OpenSearch Service.
3. Untuk pertimbangan tentang menggunakan penulisan, lihat the section called “Sumber daya lain yang didukung”.
4. Lihat metode PUT. Untuk informasi selengkapnya tentang metode GET, lihat the section called “Perbedaan API yang mencolok”. Daftar ini hanya mengacu pada operasi Elasticsearch generik yang didukung OpenSearch Layanan dan tidak menyertakan operasi yang didukung khusus plugin untuk deteksi anomali, ISM, dan sebagainya.
5. Lihat the section called “Kecilkan”.

Elasticsearch versi 6.0

Untuk Elasticsearch 6.0, OpenSearch Layanan mendukung operasi berikut.

- Semua operasi di jalur indeks (seperti */index-name* `/_forcemerge` dan */index-name* `/update/`*id*) kecuali */index-name* `/_close`
- `/_alias`
- `/_aliases`
- `/_all`
- `/_analyze`
- `/_bulk`
- `/_cat` (kecuali `/_cat/nodeattrs`)
- `/_cluster/allocation/explain`
- `/_cluster/health`
- `/_cluster/pending_tasks`
- `/_cluster/state`
- `/_cluster/stats`
- `/_count`
- `/_delete_by_query` [1]
- `/_explain`
- `/_field_caps`
- `/_field_stats`
- `/_flush`
- `/_ingest/pipeline`
- `/_mapping`
- `/_mget`
- `/_msearch`
- `/_mtermvectors`
- `/_nodes`
- `/_percolate`
- `/_plugin/kibana`
- `/_render`
- `/_rollover`
- `/_scripts` [3]
- `/_search`[2]
- `/_search profile`
- `/_shard_stores`
- `/_shrink`[5]
- `/_snapshot`
- `/_stats`
- `/_status`
- `/_tasks`
- `/_template`
- `/_update_by_query` [1]
- `/_validate`

- `/_cluster/settings` untuk beberapa properti[4]:
  - `action.auto_create_index`
  - `action.search.shard_count.limit`
  - `indices.breaker.fielddata.limit`
  - `indices.breaker.request.limit`
  - `indices.breaker.total.limit`
- `/_refresh`
- `/_reindex` [1]

1. Perubahan konfigurasi klaster mungkin mengganggu operasi ini sebelum selesai. Kami menyarankan agar Anda menggunakan operasi `/_tasks` bersama dengan operasi ini untuk memverifikasi bahwa permintaan berhasil diselesaikan.
2. HAPUS permintaan untuk `/_search/scroll` dengan badan pesan harus menentukan `"Content-Length"` di header HTTP. Kebanyakan klien menambahkan header ini secara default. Untuk menghindari masalah dengan = karakter dalam `scroll_id` nilai, gunakan badan permintaan, bukan string kueri, untuk meneruskan `scroll_id` nilai ke OpenSearch Service.
3. Untuk pertimbangan tentang menggunakan penulisan, lihat the section called “Sumber daya lain yang didukung”.
4. Lihat metode `PUT`. Untuk informasi selengkapnya tentang metode `GET`, lihat the section called “Perbedaan API yang mencolok”. Daftar ini hanya mengacu pada operasi Elasticsearch generik yang didukung OpenSearch Layanan dan tidak menyertakan operasi yang didukung khusus plugin untuk deteksi anomali, ISM, dan sebagainya.
5. Lihat the section called “Kecilkan”.

Elasticsearch versi 5.6

Untuk Elasticsearch 5.6, OpenSearch Layanan mendukung operasi berikut.

- Semua operasi di jalur indeks (seperti */index-name* / _forcemerge dan */index-name* /update/*id*) kecuali */index-name* /_close
- /_alias
- /_aliases
- /_all
- /_analyze
- /_bulk
- /_cat (kecuali /_cat/nodeattrs)
- /_cluster/allocation/explain
- /_cluster/health
- /_cluster/pending_tasks
- /_cluster/settings untuk beberapa properti[4]:
  - action.auto_create_index
  - action.search.shard_count.limit
  - indices.breaker.fielddata.limit
  - indices.breaker.request.limit
  - indices.breaker.total.limit
- /_cluster/state
- /_cluster/stats
- /_count
- /_delete_by_query [1]
- /_explain
- /_field_caps
- /_field_stats
- /_flush
- /_ingest/pipeline
- /_mapping
- /_mget
- /_msearch
- /_mtermvectors
- /_nodes
- /_percolate
- /_plugin/kibana
- /_refresh
- /_reindex [1]
- /_render
- /_rollover
- /_scripts [3]
- /_search[2]
- /_search profile
- /_shard_stores
- /_shrink[5]
- /_snapshot
- /_stats
- /_status
- /_tasks
- /_template
- /_update_by_query [1]
- /_validate

1. Perubahan konfigurasi klaster mungkin mengganggu operasi ini sebelum selesai. Kami menyarankan agar Anda menggunakan operasi /_tasks bersama dengan operasi ini untuk memverifikasi bahwa permintaan berhasil diselesaikan.

2. HAPUS permintaan untuk `/_search/scroll` dengan badan pesan harus menentukan `"Content-Length"` di header HTTP. Kebanyakan klien menambahkan header ini secara default. Untuk menghindari masalah dengan = karakter dalam `scroll_id` nilai, gunakan badan permintaan, bukan string kueri, untuk meneruskan `scroll_id` nilai ke OpenSearch Service.
3. Untuk pertimbangan tentang menggunakan penulisan, lihat the section called “Sumber daya lain yang didukung”.
4. Lihat metode PUT. Untuk informasi selengkapnya tentang metode GET, lihat the section called “Perbedaan API yang mencolok”. Daftar ini hanya mengacu pada operasi Elasticsearch generik yang didukung OpenSearch Layanan dan tidak menyertakan operasi yang didukung khusus plugin untuk deteksi anomali, ISM, dan sebagainya.
5. Lihat the section called “Kecilkan”.

Elasticsearch versi 5.5

Untuk Elasticsearch 5.5, OpenSearch Layanan mendukung operasi berikut.

- Semua operasi di jalur indeks (seperti */index-name* / `_forcemerge` dan */index-name* `/update/`*id*) kecuali */index-name* `/_close`
- `/_alias`
- `/_aliases`
- `/_all`
- `/_analyze`
- `/_bulk`
- `/_cat` (kecuali `/_cat/nodeattrs`)
- `/_cluster/allocation/explain`
- `/_cluster/health`
- `/_cluster/pending_tasks`
- `/_cluster/state`
- `/_cluster/stats`
- `/_count`
- `/_delete_by_query` [1]
- `/_explain`
- `/_field_caps`
- `/_field_stats`
- `/_flush`
- `/_ingest/pipeline`
- `/_mapping`
- `/_mget`
- `/_msearch`
- `/_mtermvectors`
- `/_nodes`
- `/_percolate`
- `/_plugin/kibana`
- `/_render`
- `/_rollover`
- `/_scripts` [3]
- `/_search`[2]
- `/_search profile`
- `/_shard_stores`
- `/_shrink`[5]
- `/_snapshot`
- `/_stats`
- `/_status`
- `/_tasks`
- `/_template`
- `/_update_by_query` [1]
- `/_validate`

- `/_cluster/settings` untuk beberapa properti[4]:
  - `action.auto_create_index`
  - `action.search.shard_count.limit`
  - `indices.breaker.fielddata.limit`
  - `indices.breaker.request.limit`
  - `indices.breaker.total.limit`
- `/_refresh`
- `/_reindex` [1]

1. Perubahan konfigurasi klaster mungkin mengganggu operasi ini sebelum selesai. Kami menyarankan agar Anda menggunakan operasi `/_tasks` bersama dengan operasi ini untuk memverifikasi bahwa permintaan berhasil diselesaikan.
2. HAPUS permintaan untuk `/_search/scroll` dengan badan pesan harus menentukan `"Content-Length"` di header HTTP. Kebanyakan klien menambahkan header ini secara default. Untuk menghindari masalah dengan = karakter dalam `scroll_id` nilai, gunakan badan permintaan, bukan string kueri, untuk meneruskan `scroll_id` nilai ke OpenSearch Service.
3. Untuk pertimbangan tentang menggunakan penulisan, lihat the section called “Sumber daya lain yang didukung”.
4. Lihat metode `PUT`. Untuk informasi selengkapnya tentang metode `GET`, lihat the section called “Perbedaan API yang mencolok”. Daftar ini hanya mengacu pada operasi Elasticsearch generik yang didukung OpenSearch Layanan dan tidak menyertakan operasi yang didukung khusus plugin untuk deteksi anomali, ISM, dan sebagainya.
5. Lihat the section called “Kecilkan”.

Elasticsearch versi 5.3

Untuk Elasticsearch 5.3, OpenSearch Layanan mendukung operasi berikut.

- Semua operasi di jalur indeks (seperti */index-name* / _forcemerge dan */index-name* /update/*id*) kecuali */index-name* /_close
- /_alias
- /_aliases
- /_all
- /_analyze
- /_bulk
- /_cat (kecuali /_cat/nodeattrs)
- /_cluster/allocation/explain
- /_cluster/health
- /_cluster/pending_tasks
- /_cluster/settings untuk beberapa properti[3]:
  - action.auto_create_index
  - action.search.shard_count.limit
  - indices.breaker.fielddata.limit
  - indices.breaker.request.limit
  - indices.breaker.total.limit
- /_cluster/state
- /_cluster/stats
- /_count
- /_delete_by_query [1]
- /_explain
- /_field_caps
- /_field_stats
- /_flush
- /_ingest/pipeline
- /_mapping
- /_mget
- /_msearch
- /_mtermvectors
- /_nodes
- /_percolate
- /_plugin/kibana
- /_refresh
- /_reindex [1]
- /_render
- /_rollover
- /_search[2]
- /_search profile
- /_shard_stores
- /_shrink[4]
- /_snapshot
- /_stats
- /_status
- /_tasks
- /_template
- /_update_by_query [1]
- /_validate

1. Perubahan konfigurasi klaster mungkin mengganggu operasi ini sebelum selesai. Kami menyarankan agar Anda menggunakan operasi /_tasks bersama dengan operasi ini untuk memverifikasi bahwa permintaan berhasil diselesaikan.

2. HAPUS permintaan untuk `/_search/scroll` dengan badan pesan harus menentukan `"Content-Length"` di header HTTP. Kebanyakan klien menambahkan header ini secara default. Untuk menghindari masalah dengan = karakter dalam `scroll_id` nilai, gunakan badan permintaan, bukan string kueri, untuk meneruskan `scroll_id` nilai ke OpenSearch Service.
3. Lihat metode PUT. Untuk informasi selengkapnya tentang metode GET, lihat the section called "Perbedaan API yang mencolok". Daftar ini hanya mengacu pada operasi Elasticsearch generik yang didukung OpenSearch Layanan dan tidak menyertakan operasi yang didukung khusus plugin untuk deteksi anomali, ISM, dan sebagainya.
4. Lihat the section called "Kecilkan".

Elasticsearch versi 5.1

Untuk Elasticsearch 5.1, OpenSearch Layanan mendukung operasi berikut.

- Semua operasi di jalur indeks (seperti */index-name* / `_forcemerge` dan */index-name* `/update/`*id*) kecuali */index-name* `/_close`
- `/_alias`
- `/_aliases`
- `/_all`
- `/_analyze`
- `/_bulk`
- `/_cat` (kecuali `/_cat/nodeattrs`)
- `/_cluster/allocation/explain`
- `/_cluster/health`
- `/_cluster/pending_tasks`
- `/_cluster/settings` untuk beberapa properti (PUT saja):
- `/_cluster/state`
- `/_cluster/stats`
- `/_count`
- `/_delete_by_query` [1]
- `/_explain`
- `/_field_caps`
- `/_field_stats`
- `/_flush`
- `/_ingest/pipeline`
- `/_mapping`
- `/_mget`
- `/_msearch`
- `/_mtermvectors`
- `/_nodes`
- `/_percolate`
- `/_plugin/kibana`
- `/_refresh`
- `/_reindex` [1]
- `/_render`
- `/_rollover`
- `/_search` [2]
- `/_search profile`
- `/_shard_stores`
- `/_shrink` [3]
- `/_snapshot`
- `/_stats`
- `/_status`
- `/_tasks`
- `/_template`
- `/_update_by_query` [1]
- `/_validate`

- `action.auto_create_index`
- `action.search.shard_count.limit`
- `indices.breaker.fielddata.limit`
- `indices.breaker.request.limit`
- `indices.breaker.total.limit`

1. Perubahan konfigurasi klaster mungkin mengganggu operasi ini sebelum selesai. Kami menyarankan agar Anda menggunakan operasi `/_tasks` bersama dengan operasi ini untuk memverifikasi bahwa permintaan berhasil diselesaikan.
2. HAPUS permintaan untuk `/_search/scroll` dengan badan pesan harus menentukan `"Content-Length"` di header HTTP. Kebanyakan klien menambahkan header ini secara default. Untuk menghindari masalah dengan = karakter dalam `scroll_id` nilai, gunakan badan permintaan, bukan string kueri, untuk meneruskan `scroll_id` nilai ke OpenSearch Service.
3. Lihat the section called “Kecilkan”.

Elasticsearch versi 2.3

Untuk Elasticsearch 2.3, OpenSearch Layanan mendukung operasi berikut.

- Semua operasi di jalur indeks (seperti `/index-name /_forcemerge` dan `/index-name /_recovery`) kecuali `/index-name /_close`
- `/_alias`
- `/_aliases`
- `/_all`
- `/_analyze`
- `/_bulk`
- `/_cluster/stats`
- `/_count`
- `/_flush`
- `/_mapping`
- `/_mget`
- `/_msearch`
- `/_nodes`
- `/_percolate`

- `/_cache/clear` (hanya indeks)
- `/_cat` (kecuali `/_cat/nodeattrs`)
- `/_cluster/health`
- `/_cluster/settings` untuk beberapa properti (PUT saja):
  - `indices.breaker.fielddata.limit`
  - `indices.breaker.request.limit`
  - `indices.breaker.total.limit`
  - `threadpool.get.queue_size`
  - `threadpool.bulk.queue_size`
  - `threadpool.index.queue_size`
  - `threadpool.percolate.queue_size`
  - `threadpool.search.queue_size`
  - `threadpool.suggest.queue_size`
- `/_plugin/kibana`
- `/_refresh`
- `/_render`
- `/_search`
- `/_snapshot`
- `/_stats`
- `/_status`
- `/_template`

Elasticsearch versi 1.5

Untuk Elasticsearch 1.5, OpenSearch Layanan mendukung operasi berikut.

- Semua operasi di jalur indeks, seperti */index-name* `/_optimize` dan */index-name* `/_warmer`, kecuali */index-name* `/_close`
- `/_alias`
- `/_aliases`
- `/_all`
- `/_analyze`
- `/_bulk`
- `/_cat`
- `/_cluster/health`
- `/_cluster/stats`
- `/_count`
- `/_flush`
- `/_mapping`
- `/_mget`
- `/_msearch`
- `/_nodes`
- `/_percolate`
- `/_plugin/kibana`
- `/_plugin/kibana3`

- `/_cluster/settings` untuk beberapa properti (PUT saja):
  - `indices.breaker.fielddata.limit`
  - `indices.breaker.request.limit`
  - `indices.breaker.total.limit`
  - `threadpool.get.queue_size`
  - `threadpool.bulk.queue_size`
  - `threadpool.index.queue_size`
  - `threadpool.percolate.queue_size`
  - `threadpool.search.queue_size`
  - `threadpool.suggest.queue_size`
- `/_plugin/migration`
- `/_refresh`
- `/_search`
- `/_snapshot`
- `/_stats`
- `/_status`
- `/_template`

Kuota OpenSearch Layanan Amazon

AWS Akun Anda memiliki kuota default, sebelumnya disebut sebagai batas, untuk setiap layanan. AWS Kecuali dinyatakan sebaliknya, setiap kuota unik untuk suatu Wilayah.

Untuk melihat kuota untuk domain dan instance OpenSearch Layanan, Amazon OpenSearch Tanpa Server, dan Amazon OpenSearch Ingestion, lihat kuota Layanan Amazon di. OpenSearch Referensi Umum AWS

Untuk melihat kuota OpenSearch Layanan di AWS Management Console, buka konsol Service Quotas. Di panel navigasi, pilih AWS layanan dan pilih OpenSearchLayanan Amazon. Untuk meminta penambahan kuota, lihat Meminta penambahan kuota di Panduan Pengguna Service Quotas.

Topik

- UltraWarm kuota penyimpanan
- Kuota ukuran volume EBS
- Kuota jaringan
- Kuota ukuran pecahan
- Kuota proses Java

- Kuota kebijakan domain

UltraWarm kuota penyimpanan

Tabel berikut mencantumkan jenis UltraWarm instance dan jumlah maksimum penyimpanan yang dapat digunakan setiap jenis. Untuk informasi lebih lanjut tentang UltraWarm, lihatthe section called "UltraWarm penyimpanan".

Jenis instans	Penyimpanan maksimum
`ultrawarm1.medium.search`	1.5 TIB
`ultrawarm1.large.search`	20 TiB

Kuota ukuran volume EBS

Tabel berikut menunjukkan ukuran minimum dan maksimum untuk volume EBS untuk setiap jenis instans yang didukung OpenSearch Layanan. Untuk informasi tentang jenis instans yang menyertakan penyimpanan instans dan detail perangkat keras tambahan, lihat harga OpenSearch Layanan Amazon.

- Jika Anda memilih penyimpanan magnetik di bawah tipe volume EBS saat membuat domain Anda, ukuran volume maksimum adalah 100 GiB untuk semua jenis instans `t2.small` kecuali `t2.medium` dan, dan semua instance Graviton (m6G, C6g, R6g, dan R6gd), yang tidak mendukung penyimpanan magnetik. Untuk ukuran maksimum yang tercantum dalam tabel berikut, pilih salah satu opsi SSD.
- Beberapa tipe instans generasi lama mencakup penyimpanan instans, sekaligus mendukung penyimpanan EBS. Jika Anda memilih penyimpanan EBS untuk salah satu tipe instans ini, volume penyimpanannya tidak bersifat aditif. Anda dapat menggunakan volume EBS atau penyimpanan instans, tidak keduanya.

Tipe instans	Ukuran minimum EBS	Ukuran EBS maksimum (gp2)	Ukuran EBS maksimum (gp3)
t2.micro.search	10 GiB	35 GiB	N/A
t2.small.search	10 GiB	35 GiB	N/A
t2.medium.search	10 GiB	35 GiB	N/A
t3.small.search	10 GiB	100 GiB	100 GiB
t3.medium.search	10 GiB	200 GiB	200 GiB
m3.medium.search	10 GiB	100 GiB	N/A
m3.large.search	10 GiB	512 GiB	N/A
m3.xlarge.search	10 GiB	512 GiB	N/A
m3.2xlarge.search	10 GiB	512 GiB	N/A
m4.large.search	10 GiB	512 GiB	N/A
m4.xlarge.search	10 GiB	1 TiB	N/A
m4.2xlarge.search	10 GiB	1.5 TIB	N/A
m4.4xlarge.search	10 GiB	1.5 TIB	N/A
m4.10xlarge.search	10 GiB	1.5 TIB	N/A
m5.large.search	10 GiB	512 GiB	1 TiB
m5.xlarge.search	10 GiB	1 TiB	2 TiB
m5.2xlarge.search	10 GiB	1.5 TIB	3 TiB
m5.4xlarge.search	10 GiB	3 TiB	6 TiB
m5.12xlarge.search	10 GiB	9 TiB	18 TiB

Tipe instans	Ukuran minimum EBS	Ukuran EBS maksimum (gp2)	Ukuran EBS maksimum (gp3)
m6g.large.search	10 GiB	512 GiB	1 TiB
m6g.xlarge.search	10 GiB	1 TiB	2 TiB
m6g.2xlarge.search	10 GiB	1.5 TIB	3 TiB
m6g.4xlarge.search	10 GiB	3 TiB	6 TiB
m6g.8xlarge.search	10 GiB	6 TiB	12 TiB
m6g.12xlarge.search	10 GiB	9 TiB	18 TiB
c4.large.search	10 GiB	100 GiB	N/A
c4.xlarge.search	10 GiB	512 GiB	N/A
c4.2xlarge.search	10 GiB	1 TiB	N/A
c4.4xlarge.search	10 GiB	1.5 TIB	N/A
c4.8xlarge.search	10 GiB	1.5 TIB	N/A
c5.large.search	10 GiB	256 GiB	256 GiB
c5.xlarge.search	10 GiB	512 GiB	512 GiB
c5.2xlarge.search	10 GiB	1 TiB	1 TiB
c5.4xlarge.search	10 GiB	1.5 TIB	1.5 TIB
c5.9xlarge.search	10 GiB	3.5 TiB	3.5 TiB
c5.18xlarge.search	10 GiB	7 TiB	7 TiB
c6g.large.search	10 GiB	256 GiB	256 GiB
c6g.xlarge.search	10 GiB	512 GiB	512 GiB

Tipe instans	Ukuran minimum EBS	Ukuran EBS maksimum (gp2)	Ukuran EBS maksimum (gp3)
c6g.2xlarge.search	10 GiB	1 TiB	1 TiB
c6g.4xlarge.search	10 GiB	1.5 TIB	1.5 TIB
c6g.8xlarge.search	10 GiB	3 TiB	3 TiB
c6g.12xlarge.search	10 GiB	4.5 TiB	4.5 TiB
r3.large.search	10 GiB	512 GiB	N/A
r3.xlarge.search	10 GiB	512 GiB	N/A
r3.2xlarge.search	10 GiB	512 GiB	N/A
r3.4xlarge.search	10 GiB	512 GiB	N/A
r3.8xlarge.search	10 GiB	512 GiB	N/A
r4.large.search	10 GiB	1 TiB	N/A
r4.xlarge.search	10 GiB	1.5 TIB	N/A
r4.2xlarge.search	10 GiB	1.5 TIB	N/A
r4.4xlarge.search	10 GiB	1.5 TIB	N/A
r4.8xlarge.search	10 GiB	1.5 TIB	N/A
r4.16xlarge.search	10 GiB	1.5 TIB	N/A
r5.large.search	10 GiB	1 TiB	2 TiB
r5.xlarge.search	10 GiB	1.5 TIB	3 TiB
r5.2xlarge.search	10 GiB	3 TiB	6 TiB
r5.4xlarge.search	10 GiB	6 TiB	12 TiB

Tipe instans	Ukuran minimum EBS	Ukuran EBS maksimum (gp2)	Ukuran EBS maksimum (gp3)
r5.12xlarge.search	10 GiB	12 TiB	24 TiB
r6g.large.search	10 GiB	1 TiB	2 TiB
r6g.xlarge.search	10 GiB	1.5 TIB	3 TiB
r6g.2xlarge.search	10 GiB	3 TiB	6 TiB
r6g.4xlarge.search	10 GiB	6 TiB	12 TiB
r6g.8xlarge.search	10 GiB	8 TiB	16 TiB
r6g.12xlarge.search	10 GiB	12 TiB	24 TiB
r6gd.large.search	N/A	N/A	N/A
r6gd.xlarge.search	N/A	N/A	N/A
r6gd.2xlarge.search	N/A	N/A	N/A
r6gd.4xlarge.search	N/A	N/A	N/A
r6gd.8xlarge.search	N/A	N/A	N/A
r6gd.12xlarge.search	N/A	N/A	N/A
r6gd.16xlarge.search	N/A	N/A	N/A
i2.xlarge.search	10 GiB	512 GiB	N/A
i2.2xlarge.search	10 GiB	512 GiB	N/A
i3.large.search	N/A	N/A	N/A
i3.xlarge.search	N/A	N/A	N/A
i3.2xlarge.search	N/A	N/A	N/A

Tipe instans	Ukuran minimum EBS	Ukuran EBS maksimum (gp2)	Ukuran EBS maksimum (gp3)
i3.4xlarge.search	N/A	N/A	N/A
i3.8xlarge.search	N/A	N/A	N/A
i3.16xlarge.search	N/A	N/A	N/A
or1.medium.search	20 GiB	N/A	768 GiB
or1.large.search	20 GiB	N/A	1532 GiB
or1.xlarge.search	20 GiB	N/A	3 TiB
or1.2xlarge.search	20 GiB	N/A	6 TiB
or1.4xlarge.search	20 GiB	N/A	12 TiB
or1.8xlarge.search	20 GiB	N/A	16 TiB
or1.12xlarge.search	20 GiB	N/A	24 TiB
or1.16xlarge.search	20 GiB	N/A	36 TiB
im4gn.large.search	N/A	N/A	N/A
im4gn.xlarge.search	N/A	N/A	N/A
im4gn.2xlarge.search	N/A	N/A	N/A
im4gn.4xlarge.search	N/A	N/A	N/A
im4gn.8xlarge.search	N/A	N/A	N/A
im4gn.16xlarge.search	N/A	N/A	N/A

Kuota jaringan

Tabel berikut menunjukkan ukuran maksimum muatan permintaan HTTP.

Tipe instans	Ukuran maksimum muatan permintaan HTTP
t2.micro.search	10 MiB
t2.small.search	10 MiB
t2.medium.search	10 MiB
t3.small.search	10 MiB
t3.medium.search	10 MiB
m3.medium.search	10 MiB
m3.large.search	10 MiB
m3.xlarge.search	100 MiB
m3.2xlarge.search	100 MiB
m4.large.search	10 MiB
m4.xlarge.search	100 MiB
m4.2xlarge.search	100 MiB
m4.4xlarge.search	100 MiB
m4.10xlarge.search	100 MiB
m5.large.search	10 MiB
m5.xlarge.search	100 MiB
m5.2xlarge.search	100 MiB

Tipe instans	Ukuran maksimum muatan permintaan HTTP
m5.4xlarge.search	100 MiB
m5.12xlarge.search	100 MiB
m6g.large.search	10 MiB
m6g.xlarge.search	100 MiB
m6g.2xlarge.search	100 MiB
m6g.4xlarge.search	100 MiB
m6g.8xlarge.search	100 MiB
m6g.12xlarge.search	100 MiB
c4.large.search	10 MiB
c4.xlarge.search	100 MiB
c4.2xlarge.search	100 MiB
c4.4xlarge.search	100 MiB
c4.8xlarge.search	100 MiB
c5.large.search	10 MiB
c5.xlarge.search	100 MiB
c5.2xlarge.search	100 MiB
c5.4xlarge.search	100 MiB

Tipe instans	Ukuran maksimum muatan permintaan HTTP
c5.9xlarge.search	100 MiB
c5.18xlarge.search	100 MiB
c6g.large.search	10 MiB
c6g.xlarge.search	100 MiB
c6g.2xlarge.search	100 MiB
c6g.4xlarge.search	100 MiB
c6g.8xlarge.search	100 MiB
c6g.12xlarge.search	100 MiB
r3.large.search	10 MiB
r3.xlarge.search	100 MiB
r3.2xlarge.search	100 MiB
r3.4xlarge.search	100 MiB
r3.8xlarge.search	100 MiB
r4.large.search	100 MiB
r4.xlarge.search	100 MiB
r4.2xlarge.search	100 MiB
r4.4xlarge.search	100 MiB

Tipe instans	Ukuran maksimum muatan permintaan HTTP
r4.8xlarge.search	100 MiB
r4.16xlarge.search	100 MiB
r5.large.search	100 MiB
r5.xlarge.search	100 MiB
r5.2xlarge.search	100 MiB
r5.4xlarge.search	100 MiB
r5.12xlarge.search	100 MiB
r6g.large.search	100 MiB
r6g.xlarge.search	100 MiB
r6g.2xlarge.search	100 MiB
r6g.4xlarge.search	100 MiB
r6g.8xlarge.search	100 MiB
r6g.12xlarge.search	100 MiB
r6gd.large.search	100 MiB
r6gd.xlarge.search	100 MiB
r6gd.2xlarge.search	100 MiB

Tipe instans	Ukuran maksimum muatan permintaan HTTP
r6gd.4xlarge.search	100 MiB
r6gd.8xlarge.search	100 MiB
r6gd.12xlarge.search	100 MiB
r6gd.16xlarge.search	100 MiB
i2.xlarge.search	100 MiB
i2.2xlarge.search	100 MiB
i3.large.search	100 MiB
i3.xlarge.search	100 MiB
i3.2xlarge.search	100 MiB
i3.4xlarge.search	100 MiB
i3.8xlarge.search	100 MiB
i3.16xlarge.search	100 MiB
or1.medium.search	10 MiB
or1.large.search	100 MiB
or1.xlarge.search	100 MiB
or1.2xlarge.search	100 MiB

Tipe instans	Ukuran maksimum muatan permintaan HTTP
or1.4xlarge.search	100 MiB
or1.8xlarge.search	100 MiB
or1.12xlarge.search	100 MiB
or1.16xlarge.search	100 MiB
im4gn.large.search	100 MiB
im4gn.xlarge.search	100 MiB
im4gn.2xlarge.search	100 MiB
im4gn.4xlarge.search	100 MiB
im4gn.8xlarge.search	100 MiB
im4gn.16xlarge.search	100 MiB

Kuota ukuran pecahan

Bagian berikut mencantumkan ukuran pecahan maksimum untuk berbagai keluarga instance.

Jenis instans	Multi-AZ tanpa Siaga	Multi-AZ dengan Siaga
R5, C5, M5	N/A	65 GiB
I3	N/A	65 GiB
R6g, C6g, M6g, R6gd	N/A	65 GiB
ATAU1	100 GiB	65 GiB
Im4gn	N/A	65 GiB

Untuk meminta peningkatan kuota, hubungi AWS Support.

Kuota proses Java

OpenSearch Layanan membatasi proses Java ke ukuran heap 32 GiB. Pengguna lanjutan dapat menentukan persentase tumpukan yang digunakan untuk data lapangan. Untuk informasi selengkapnya, lihat the section called “Pengaturan cluster lanjutan” dan the section called “JVM OutOfMemoryError”.

Kuota kebijakan domain

OpenSearch Layanan membatasi kebijakan akses pada domain hingga 100 KiB.

Instans yang Disimpan di Amazon OpenSearch Layanan

Instans Cadangan (RI) di Amazon OpenSearch Layanan menawarkan diskon yang signifikan dibandingkan dengan Instans Sesuai Permintaan standar. Instans itu sendiri identik; RI hanya diskon penagihan yang diterapkan Instans Sesuai Permintaan di akun Anda. Untuk aplikasi jangka panjang dengan penggunaan yang dapat diprediksi, RI dapat memberikan penghematan yang cukup besar seiring waktu.

OpenSearch RI Layanan membutuhkan jangka waktu satu atau tiga tahun dan memiliki tiga pilihan pembayaran yang mempengaruhi tingkat discount:

- Tidak ada Pembayaran di Muka - Anda tidak perlu membayar di muka. Anda membayar tarif per jam diskon untuk setiap jam dalam jangka waktu tersebut.

- Pembayaran di Muka Sebagian — Anda membayar sebagian biaya di muka, dan Anda membayar tarif per jam diskon untuk setiap jam dalam jangka waktu tersebut.
- Pembayaran Di Muka Penuh - Anda membayar keseluruhan biaya di muka. Anda tidak membayar tarif per jam untuk jangka waktu tersebut.

Secara umum, pembayaran di muka yang lebih besar berarti diskon yang lebih besar. Anda tidak dapat membatalkan Instans Cadangan—saat Anda memesannya, Anda berkomitmen untuk membayar keseluruhan jangka waktu—dan pembayaran di muka tidak dapat dikembalikan.

RI tidak fleksibel; mereka hanya berlaku untuk tipe instans yang persis yang Anda pesan. Misalnya, reservasi untuk delapan instans `c5.2xlarge.search` tidak berlaku untuk enam belas instans `c5.xlarge.search` atau empat instans `c5.4xlarge.search`. Untuk rincian selengkapnya, lihatAmazon OpenSearch Harga layanandanFAQ.

Topik

- Membeli Instans Cadangan (konsol)
- Membeli Instans Cadangan (AWS CLI)
- Membeli Instans Cadangan (AWS SDK)
- Memeriksa biaya

Membeli Instans Cadangan (konsol)

Konsol ini memungkinkan Anda melihat Instans Cadangan yang sudah ada dan membeli yang baru.

Untuk membeli reservasi

1. Masuk ke https://aws.amazon.com, kemudian pilih Masuk ke Konsol.
2. Di bawahAnalitik, pilihAmazon OpenSearch Layanan.
3. PilihSewa Instansdari panel navigasi.

   Pada halaman ini, Anda dapat melihat reservasi yang sudah ada. Jika Anda memiliki banyak reservasi, Anda dapat filter reservasi untuk lebih mudah mengidentifikasi dan melihat reservasi tertentu.

> **Tip**
>
> Jika Anda tidak melihatSewa Instanslink,membuat domaindi dalamWilayah AWS.

4. PilihInstans yang Disimpan.
5. Berikan nama yang unik dan deskriptif.
6. Pilih tipe instans. Untuk panduan, lihat the section called “Mengukur domain”.
7. Pilih panjang jangka waktu dan pilihan pembayaran. Tinjau detail pembayaran dengan seksama.
8. Pilih Selanjutnya.
9. Tinjau ringkasan pembelian dengan hati-hati. Pembelian Instans Cadangan tidak dapat dikembalikan.
10. PilihOrder.

Membeli Instans Cadangan (AWS CLI)

AWS CLI memiliki perintah untuk melihat penawaran, membeli reservasi, dan melihat reservasi Anda. Perintah berikut dan respon sampel menunjukkan penawaran untuk yang diberikanWilayah AWS:

```
aws opensearch describe-reserved-instance-offerings --region us-east-1
{
  "ReservedInstanceOfferings": [
    {
      "FixedPrice": x,
      "ReservedInstanceOfferingId": "1a2a3a4a5-1a2a-3a4a-5a6a-1a2a3a4a5a6a",
      "RecurringCharges": [
        {
          "RecurringChargeAmount": y,
          "RecurringChargeFrequency": "Hourly"
        }
      ],
      "UsagePrice": 0.0,
      "PaymentOption": "PARTIAL_UPFRONT",
      "Duration": 31536000,
      "InstanceType": "m4.2xlarge.search",
      "CurrencyCode": "USD"
    }
  ]
```

```
}
```

Untuk penjelasan tentang setiap nilai kembali, lihat tabel berikut.

Bidang	Deskripsi
`FixedPrice`	Biaya di muka reservasi.
`ReservedInstanceOfferingId`	ID penawaran. Catat nilai ini jika Anda ingin memesan penawaran.
`RecurringCharges`	Tarif per jam untuk reservasi.
`UsagePrice`	Bidang warisan. Untuk OpenSearch Layanan, nilai ini selalu 0.
`PaymentOption`	Tanpa Biaya di Muka, Sebagian, atau Semua Di Muka.
`Duration`	Panjang jangka waktu dalam hitungan detik: • 31536000 detik adalah satu tahun. • 94608000 detik adalah tiga tahun.
`InstanceType`	Tipe instans untuk reservasi. Untuk informasi tentang sumber daya perangkat keras yang dialokasikan untuk setiap tipe instans, lihatAmazon OpenSearch Harga layanan.
`CurrencyCode`	Mata uang untuk `FixedPrice` dan `RecurringChargeAmount` .

Contoh berikutnya ini membeli reservasi:

```
aws opensearch purchase-reserved-instance-offering --reserved-instance-offering-
id 1a2a3a4a5-1a2a-3a4a-5a6a-1a2a3a4a5a6a --reservation-name my-reservation --instance-
count 3 --region us-east-1
{
  "ReservationName": "my-reservation",
```

```
  "ReservedInstanceId": "9a8a7a6a-5a4a-3a2a-1a0a-9a8a7a6a5a4a"
}
```

Akhirnya, Anda dapat mencantumkan reservasi Anda untuk Wilayah yang diberikan dengan menggunakan contoh berikut:

```
aws opensearch describe-reserved-instances --region us-east-1
{
  "ReservedInstances": [
    {
      "FixedPrice": x,
      "ReservedInstanceOfferingId": "1a2a3a4a5-1a2a-3a4a-5a6a-1a2a3a4a5a6a",
      "ReservationName": "my-reservation",
      "PaymentOption": "PARTIAL_UPFRONT",
      "UsagePrice": 0.0,
      "ReservedInstanceId": "9a8a7a6a-5a4a-3a2a-1a0a-9a8a7a6a5a4a",
      "RecurringCharges": [
        {
          "RecurringChargeAmount": y,
          "RecurringChargeFrequency": "Hourly"
        }
      ],
      "State": "payment-pending",
      "StartTime": 1522872571.229,
      "InstanceCount": 3,
      "Duration": 31536000,
      "InstanceType": "m4.2xlarge.search",
      "CurrencyCode": "USD"
    }
  ]
}
```

> **Note**
>
> `StartTime` adalah Unix waktu jangka waktu, yang merupakan jumlah detik yang telah berlalu sejak tengah malam UTC 1 Januari 1970. Sebagai contoh, 1522872571 waktu jangka waktu adalah 20:09:31 UTC dari 4 April 2018. Anda bisa menggunakan konverter online.

Untuk mempelajari selengkapnya tentang perintah yang digunakan dalam contoh sebelumnya, lihat AWS CLI Referensi Perintah.

Membeli Instans Cadangan (AWS SDK)

KlasterAWSSDK (kecuali SDK Android dan iOS) mendukung semua operasi yang didefinisikan dalamAmazon OpenSearch Referensi API, termasuk berikut ini:

- `DescribeReservedInstanceOfferings`
- `PurchaseReservedInstanceOffering`
- `DescribeReservedInstances`

Contoh script ini menggunakanOpenSearchServiceklien Python tingkat rendah dariAWS SDK for Python (Boto3)untuk membeli Instans Cadangan. Anda harus memberikan nilai untuk`instance_type`.

```
import boto3
from botocore.config import Config

# Build the client using the default credential configuration.
# You can use the CLI and run 'aws configure' to set access key, secret
# key, and default region.

my_config = Config(
    # Optionally lets you specify a region other than your default.
    region_name='us-east-1'
)

client = boto3.client('opensearch', config=my_config)

instance_type = '' # e.g. m4.2xlarge.search


def describe_RI_offerings(client):
    """Gets the Reserved Instance offerings for this account"""

    response = client.describe_reserved_instance_offerings()
    offerings = (response['ReservedInstanceOfferings'])
    return offerings


def check_instance(offering):
    """Returns True if instance type is the one you specified above"""
```

```
    if offering['InstanceType'] == instance_type:
        return True

    return False


def get_instance_id():
    """Iterates through the available offerings to find the ID of the one you
 specified"""

    instance_type_iterator = filter(
        check_instance, describe_RI_offerings(client))
    offering = list(instance_type_iterator)
    id = offering[0]['ReservedInstanceOfferingId']
    return id


def purchase_RI_offering(client):
    """Purchase Reserved Instances"""

    response = client.purchase_reserved_instance_offering(
        ReservedInstanceOfferingId = get_instance_id(),
        ReservationName = 'my-reservation',
        InstanceCount = 1
    )
    print('Purchased reserved instance offering of type ' + instance_type)
    print(response)


def main():
    """Purchase Reserved Instances"""
    purchase_RI_offering(client)
```

Untuk informasi selengkapnya tentang menginstal dan menggunakan SDK AWS, lihat AWS Kit Pengembangan Perangkat Lunak.

Memeriksa biaya

Cost Explorer adalah alat gratis yang dapat Anda gunakan untuk melihat data pengeluaran Anda selama 13 bulan terakhir. Menganalisis data ini membantu Anda mengidentifikasi tren dan memahami apakah RI sesuai dengan kasus penggunaan Anda. Jika Anda sudah memiliki RI, Anda dapat mengelompokkan berdasarkan Opsi Pembelian dan Tampilkan biaya yang diamortisasi untuk

membandingkan pengeluaran tersebut dengan pengeluaran Anda untuk Instans Sesuai Permintaan. Anda juga dapat mengatur anggaran penggunaan untuk memastikan Anda mengambil keuntungan penuh dari pemesanan Anda. Untuk informasi selengkapnya, lihat Menganalisis Biaya Anda dengan Cost Explorer di AWS Billing Panduan Pengguna.

Sumber daya lain yang didukung di Amazon OpenSearch Service

Topik ini menjelaskan sumber daya tambahan yang didukung Amazon OpenSearch Service.

bootstrap.memory_lock

OpenSearch Layanan memungkinkan `bootstrap.memory_lock`opensearch.yml, yang mengunci memori JVM dan mencegah sistem operasi menukarnya ke disk. Hal ini berlaku untuk semua tipe instans didukung kecuali untuk berikut ini:

- `t2.micro.search`
- `t2.small.search`
- `t2.medium.search`
- `t3.small.search`
- `t3.medium.search`

Modul Scripting

OpenSearch Layanan mendukung scripting untuk Elasticsearch 5. x dan domain selanjutnya. Ini tidak mendukung penulisan untuk 1.5 atau 2.3.

Opsi penulisan yang didukung mencakup hal berikut:

- Painless
- Ekspresi Lucene
- Mustache

Untuk domain Elasticsearch 5.5 dan yang lebih baru, dan semua OpenSearch domain, OpenSearch Service mendukung skrip tersimpan menggunakan endpoint. `_scripts` Domain Elasticsearch 5.3 dan 5.1 hanya mendukung penulisan sebaris.

Transportasi TLS

OpenSearch Layanan mendukung HTTP pada port 80 dan HTTPS melalui port 443, tetapi tidak mendukung transportasi TLS.

Tutorial OpenSearch Layanan Amazon

Bab ini mencakup beberapa start-to-finish tutorial untuk bekerja dengan Amazon OpenSearch Service, termasuk bagaimana untuk bermigrasi ke layanan, membangun aplikasi pencarian sederhana, dan membuat visualisasi di OpenSearch dasbor.

Topik

Tutorial: Membuat dan mencari dokumen di Amazon OpenSearch Service

Dalam tutorial ini, Anda belajar cara membuat dan mencari dokumen di Amazon OpenSearch Service. Anda menambahkan data ke indeks dalam bentuk dokumen JSON. OpenSearch Layanan membuat indeks di sekitar dokumen pertama yang Anda tambahkan.

Tutorial ini menjelaskan cara membuat permintaan HTTP untuk membuat dokumen, secara otomatis menghasilkan ID untuk dokumen, dan melakukan pencarian dasar dan lanjutan pada dokumen Anda.

> **Note**
>
> Tutorial ini menggunakan domain dengan akses terbuka. Untuk tingkat keamanan tertinggi, kami sarankan Anda menempatkan domain Anda di dalam virtual private cloud (VPC).

Prasyarat

Tutorial ini memiliki prasyarat berikut ini:

- Anda harus memilikiAkun AWS.
- Anda harus memiliki domain OpenSearch Layanan yang aktif.

Menambahkan dokumen ke indeks

Untuk menambahkan dokumen ke indeks, Anda dapat menggunakan alat HTTP apa pun, seperti Postman, cURL, atau OpenSearch konsol Dasbor. Contoh-contoh ini mengasumsikan bahwa Anda menggunakan konsol pengembang di OpenSearch Dasbor. Jika Anda menggunakan alat yang berbeda, sesuaikan dengan memberikan URL lengkap dan kredensialnya, jika perlu.

Untuk menambahkan dokumen ke indeks

1. Arahkan ke URL OpenSearch Dasbor untuk domain Anda. Anda dapat menemukan URL di dasbor domain di konsol OpenSearch Layanan. URL mengikuti format ini:

   ```
   domain-endpoint/_dashboards/
   ```

2. Masuk menggunakan nama pengguna dan kata sandi utama Anda.
3. Buka panel navigasi kiri dan pilih Dev Tools.
4. Kata kerja HTTP untuk membuat sumber daya baru adalah PUT, yang Anda gunakan untuk membuat dokumen dan indeks baru. Masukkan perintah berikut di konsol:

   ```
   PUT fruit/_doc/1
   {
     "name":"strawberry",
     "color":"red"
   }
   ```

   PUTPermintaan membuat indeks bernama buah dan menambahkan satu dokumen ke indeks dengan ID 1. Ini menghasilkan respons berikut:

   ```
   {
     "_index" : "fruit",
     "_type" : "_doc",
     "_id" : "1",
     "_version" : 1,
     "result" : "created",
     "_shards" : {
       "total" : 2,
       "successful" : 2,
       "failed" : 0
     },
     "_seq_no" : 0,
   ```

```
  "_primary_term" : 1
}
```

Membuat ID yang dihasilkan secara otomatis

OpenSearch Layanan dapat secara otomatis menghasilkan ID untuk dokumen Anda. Perintah untuk menghasilkan ID menggunakan permintaan POST alih-alih permintaan PUT, dan tidak memerlukan ID dokumen (dibandingkan dengan permintaan sebelumnya).

Masukkan permintaan berikut di konsol pengembang:

```
POST veggies/_doc
{
  "name":"beet",
  "color":"red",
  "classification":"root"
}
```

Permintaan ini membuat indeks bernama sayuran dan menambahkan dokumen ke indeks. Ini menghasilkan respons berikut:

```
{
  "_index" : "veggies",
  "_type" : "_doc",
  "_id" : "3WgyS4IB5DLqbRIvLxtF",
  "_version" : 1,
  "result" : "created",
  "_shards" : {
    "total" : 2,
    "successful" : 2,
    "failed" : 0
  },
  "_seq_no" : 0,
  "_primary_term" : 1
}
```

Perhatikan bahwa `_id` bidang tambahan dalam respons, yang menunjukkan bahwa ID dibuat secara otomatis.

> **Note**
>
> Anda tidak memberikan apa pun setelah `_doc` di URL, di mana ID biasanya pergi. Karena Anda membuat dokumen dengan ID yang dihasilkan, Anda belum menyediakannya. Itu dicadangkan untuk pembaruan.

Memperbarui dokumen dengan perintah POST

Untuk memperbarui dokumen, Anda menggunakan `POST` perintah HTTP dengan nomor ID.

Pertama, buat dokumen dengan ID`42`:

```
POST fruits/_doc/42
{
  "name":"banana",
  "color":"yellow"
}
```

Kemudian gunakan ID itu untuk memperbarui dokumen:

```
POST fruits/_doc/42
{
  "name":"banana",
  "color":"yellow",
  "classification":"berries"
}
```

Perintah ini memperbarui dokumen dengan bidang baru`classification`. Ini menghasilkan respons berikut:

```
{
  "_index" : "fruits",
  "_type" : "_doc",
  "_id" : "42",
  "_version" : 2,
  "result" : "updated",
  "_shards" : {
    "total" : 2,
    "successful" : 2,
    "failed" : 0
```

```
  },
  "_seq_no" : 1,
  "_primary_term" : 1
}
```

> **Note**
>
> Jika Anda mencoba memperbarui dokumen yang tidak ada, OpenSearch Layanan membuat dokumen.

Melakukan tindakan massal

Anda dapat menggunakan operasi `POST _bulk` API untuk melakukan beberapa tindakan pada satu atau beberapa indeks dalam satu permintaan. Perintah tindakan massal mengambil format berikut:

```
POST /_bulk
<action_meta>\n
<action_data>\n
<action_meta>\n
<action_data>\n
```

Setiap tindakan membutuhkan dua baris JSON. Pertama, Anda memberikan deskripsi tindakan atau metadata. Pada baris berikutnya, Anda memberikan data. Setiap bagian dipisahkan oleh baris baru (\n). Deskripsi tindakan untuk sisipan mungkin terlihat seperti ini:

```
{ "create" : { "_index" : "veggies", "_type" : "_doc", "_id" : "7" } }
```

Dan baris berikutnya yang berisi data mungkin terlihat seperti ini:

```
{ "name":"kale", "color":"green", "classification":"leafy-green" }
```

Secara keseluruhan, metadata dan data mewakili satu tindakan dalam operasi massal. Anda dapat melakukan banyak operasi dalam satu permintaan, seperti ini:

```
POST /_bulk
{ "create" : { "_index" : "veggies", "_id" : "35" } }
{ "name":"kale", "color":"green", "classification":"leafy-green" }
{ "create" : { "_index" : "veggies", "_id" : "36" } }
{ "name":"spinach", "color":"green", "classification":"leafy-green" }
```

```
{ "create" : { "_index" : "veggies", "_id" : "37" } }
{ "name":"arugula", "color":"green", "classification":"leafy-green" }
{ "create" : { "_index" : "veggies", "_id" : "38" } }
{ "name":"endive", "color":"green", "classification":"leafy-green" }
{ "create" : { "_index" : "veggies", "_id" : "39" } }
{ "name":"lettuce", "color":"green", "classification":"leafy-green" }
{ "delete" : { "_index" : "vegetables", "_id" : "1" } }
```

Perhatikan bahwa tindakan terakhir adalah a`delete`. Tidak ada data yang mengikuti `delete` tindakan tersebut.

Mencari dokumen

Sekarang data ada di cluster Anda, Anda dapat mencarinya. Misalnya, Anda mungkin ingin mencari semua sayuran akar, atau menghitung semua sayuran berdaun hijau, atau menemukan jumlah kesalahan yang dicatat per jam.

Pencarian dasar

Pencarian dasar terlihat seperti ini:

```
GET veggies/_search?q=name:l*
```

Permintaan menghasilkan respons JSON yang berisi dokumen selada.

Pencarian lanjutan

Anda dapat melakukan penelusuran yang lebih maju dengan memberikan opsi kueri sebagai JSON di badan permintaan:

```
GET veggies/_search
{
  "query": {
    "term": {
      "name": "lettuce"
    }
  }
}
```

Contoh ini juga menghasilkan respons JSON dengan dokumen selada.

Penyortiran

Anda dapat melakukan lebih banyak jenis kueri ini menggunakan penyortiran. Pertama, Anda perlu membuat ulang indeks, karena pemetaan bidang otomatis memilih jenis yang tidak dapat diurutkan secara default. Kirim permintaan berikut untuk menghapus dan membuat ulang indeks:

```
DELETE /veggies

PUT /veggies
{
   "mappings":{
      "properties":{
         "name":{
            "type":"keyword"
         },
         "color":{
            "type":"keyword"
         },
         "classification":{
            "type":"keyword"
         }
      }
   }
}
```

Kemudian isi kembali indeks dengan data:

```
POST /_bulk
{ "create" : { "_index" : "veggies", "_id" : "7"  } }
{ "name":"kale", "color":"green", "classification":"leafy-green" }
{ "create" : { "_index" : "veggies", "_id" : "8" } }
{ "name":"spinach", "color":"green", "classification":"leafy-green" }
{ "create" : { "_index" : "veggies", "_id" : "9" } }
{ "name":"arugula", "color":"green", "classification":"leafy-green" }
{ "create" : { "_index" : "veggies", "_id" : "10" } }
{ "name":"endive", "color":"green", "classification":"leafy-green" }
{ "create" : { "_index" : "veggies", "_id" : "11" } }
{ "name":"lettuce", "color":"green", "classification":"leafy-green" }
```

Sekarang Anda dapat mencari dengan semacam. Permintaan ini menambahkan pengurutan naik berdasarkan klasifikasi:

```
GET veggies/_search
{
```

```
  "query" : {
    "term": { "color": "green" }
  },
  "sort" : [
      "classification"
  ]
}
```

Sumber daya terkait

Untuk informasi lebih lanjut, lihat sumber daya berikut:

- Mulai
- Mengindeks data
- Pencarian data

Tutorial: Migrasi ke AmazonOpenSearchLayanan

Snapshot indeks adalah cara populer untuk bermigrasi dari yang dikelola sendiriOpenSearchatau klaster Elasticsearch lama ke AmazonOpenSearchLayanan. Secara umum, prosesnya terdiri dari langkah-langkah berikut:

1. Ambil snapshot dari klaster yang ada, dan upload snapshot ke bucket Amazon S3.
2. Buat sebuahOpenSearchDomain layanan.
3. MemberikanOpenSearchIzin layanan untuk mengakses bucket, dan memastikan Anda memiliki izin untuk bekerja dengan snapshot.
4. Kembalikan snapshot padaOpenSearchDomain layanan.

Panduan ini menyediakan langkah-langkah yang lebih detail dan opsi lain, jika berlaku.

Mengambil dan mengunggah snapshot

Meskipun Anda dapat menggunakanrepositori-s3plugin untuk mengambil snapshot langsung ke S3, Anda harus menginstal plugin pada setiap node, men-tweak`opensearch.yml`(atau`elasticsearch.yml`jika menggunakan klaster Elasticsearch), restart setiap node, tambahkanAWSkredensi, dan akhirnya ambil snapshot. Plugin adalah pilihan yang sangat tepat untuk penggunaan berkelanjutan atau untuk migrasi klaster yang berukuran besar.

Untuk klaster yang berukuran kecil, pendekatan satu kali bertujuan mengambil snapshot sistem file bersama lalu menggunakan AWS CLI untuk mengunggahnya ke S3. Jika sudah memiliki snapshot, Anda dapat lompat ke langkah 4.

Untuk mengambil snapshot dan mengunggahnya ke Amazon S3

1. Tambahkan`path.repo`pengaturan ke`opensearch.yml`(atau`Elasticsearch.yml`) pada semua node, dan kemudian restart setiap node.

   ```
   path.repo: ["/my/shared/directory/snapshots"]
   ```

2. Daftarkanrepositori snapshot, yang diperlukan sebelum Anda mengambil snapshot. Repositori hanyalah lokasi penyimpanan: sistem file bersama, Amazon S3, Hadoop Distributed File System (HDFS), dll. Dalam kasus ini, kita akan menggunakan sistem file bersama (“fs”):

   ```
   PUT _snapshot/my-snapshot-repo-name
   {
     "type": "fs",
     "settings": {
       "location": "/my/shared/directory/snapshots"
     }
   }
   ```

3. Mengambil snapshot:

   ```
   PUT _snapshot/my-snapshot-repo-name/my-snapshot-name
   {
     "indices": "migration-index1,migration-index2,other-indices-*",
     "include_global_state": false
   }
   ```

4. Instal AWS CLI, lalu jalankan `aws configure` untuk menambahkan kredensial Anda.

5. Buka direktori snapshot. Kemudian jalankan perintah berikut untuk membuat bucket S3 yang baru dan mengunggah konten direktori snapshot ke bucket tersebut:

   ```
   aws s3 mb s3://bucket-name --region us-west-2
   aws s3 sync . s3://bucket-name --sse AES256
   ```

   Operasi ini dapat berlangsung beberapa saat bergantung pada ukuran snapshot dan kecepatan koneksi internet Anda.

Membuat domain

Meskipun konsol merupakan cara termudah untuk membuat domain, dalam hal ini, Anda sudah memiliki terminal terbuka dan AWS CLI terinstal. Ubah perintah berikut ini untuk membuat domain yang sesuai dengan kebutuhan Anda:

```
aws opensearch create-domain \
  --domain-name migration-domain \
  --engine-version OpenSearch_1.0 \
  --cluster-config InstanceType=c5.large.search,InstanceCount=2 \
  --ebs-options EBSEnabled=true,VolumeType=gp2,VolumeSize=100 \
  --node-to-node-encryption-options Enabled=true \
  --encryption-at-rest-options Enabled=true \
  --domain-endpoint-options EnforceHTTPS=true,TLSSecurityPolicy=Policy-Min-
TLS-1-2-2019-07 \
  --advanced-security-options
 Enabled=true,InternalUserDatabaseEnabled=true,MasterUserOptions='{MasterUserName=master-
user,MasterUserPassword=master-user-password}' \
  --access-policies '{"Version":"2012-10-17","Statement":
[{"Effect":"Allow","Principal":{"AWS":["*"]},"Action":
["es:ESHttp*"],"Resource":"arn:aws:es:us-west-2:123456789012:domain/migration-domain/
*"}]}' \
  --region us-west-2
```

Seperti apa adanya, perintah tersebut membuat domain yang dapat diakses oleh internet dengan dua simpul data, masing-masing dengan kapasitas penyimpanan 100GiB. Hal ini juga memungkinkan kontrol akses detail dengan autentikasi basic HTTP dan semua pengaturan enkripsi. GunakanOpenSearchKonsol layanan jika Anda memerlukan konfigurasi keamanan yang lebih canggih, seperti VPC.

Sebelum mengeluarkan perintah, ubah nama domain, kredensial pengguna utama, dan nomor akun. Tentukan yang samaWilayah AWSyang Anda gunakan untuk bucket S3 danOpenSearch/Versi Elasticsearch yang kompatibel dengan snapshot Anda.

> **Important**
>
> Snapshot hanya kompatibel dengan versi terbaru, dan hanya dengan satu versi utama. Misalnya, Anda tidak dapat memulihkan snapshot dariOpenSearch1.xcluster pada Elasticsearch 7.xcluster, hanyaOpenSearch1.xatau 2.xcluster. Versi minor juga penting. Anda tidak dapat memulihkan snapshot dari kluster 5.3.3 yang dikelola sendiri pada

5.3.2OpenSearchDomain layanan. Sebaiknya pilih versi terbaruOpenSearchatau Elasticsearch yang didukung snapshot Anda. Untuk tabel versi yang kompatibel, lihatthe section called “Menggunakan snapshot untuk memigrasi data”.

Memberikan izin untuk mengakses bucket S3

Di konsol AWS Identity and Access Management (IAM), buat peran menggunakan izin dan hubungan kepercayaan berikut. Saat membuat peran, pilih S3 sebagai Layanan AWS. Beri nama `OpenSearchSnapshotRole` pada peran agar mudah ditemukan.

Izin

```
{
  "Version": "2012-10-17",
  "Statement": [{
      "Action": [
        "s3:ListBucket"
      ],
      "Effect": "Allow",
      "Resource": [
        "arn:aws:s3:::bucket-name"
      ]
    },
    {
      "Action": [
        "s3:GetObject",
        "s3:PutObject",
        "s3:DeleteObject"
      ],
      "Effect": "Allow",
      "Resource": [
        "arn:aws:s3:::bucket-name/*"
      ]
    }
  ]
}
```

Hubungan kepercayaan

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

```
  "Statement": [{
      "Effect": "Allow",
      "Principal": {
        "Service": "es.amazonaws.com"
      },
      "Action": "sts:AssumeRole"
    }
  ]
}
```

Kemudian berikan izin peran IAM pribadi Anda untuk diasumsikan`OpenSearchSnapshotRole`. Membuat kebijakan berikut dan menyematkannya ke identitas Anda:

Izin

```
{
  "Version": "2012-10-17",
  "Statement": [{
      "Effect": "Allow",
      "Action": "iam:PassRole",
      "Resource": "arn:aws:iam::123456789012:role/OpenSearchSnapshotRole"
    }
  ]
}
```

Memetakan peran snapshot diOpenSearchDasbor (jika menggunakan kontrol akses berbutir halus)

Jika Anda mengaktifkanKontrol akses detail, meskipun menggunakan autentikasi basic HTTP untuk semua tujuan lain, Anda harus memetakan peran `manage_snapshots` ke IAM role agar dapat bekerja dengan snapshot.

Memberikan izin ke identitas Anda agar dapat menggunakan snapshot

1. Masuk ke Dasbor menggunakan kredensi pengguna utama yang Anda tentukan saat Anda membuatOpenSearchDomain layanan. Anda dapat menemukan URL Dasbor diOpenSearchKonsol layanan. URL ini menggunakan bentuk `https://domain-endpoint/_dashboards/`.
2. Dari menu utama, pilih Keamanan, Peran, lalu pilih peran manage_snapshots.
3. Pilih Pengguna yang Dipetakan, Kelola pemetaan.

4. Tambahkan ARN domain peran IAM pribadi Anda di bidang yang sesuai. ARN mengambil salah satu format berikut:

```
arn:aws:iam::123456789123:user/user-name
```

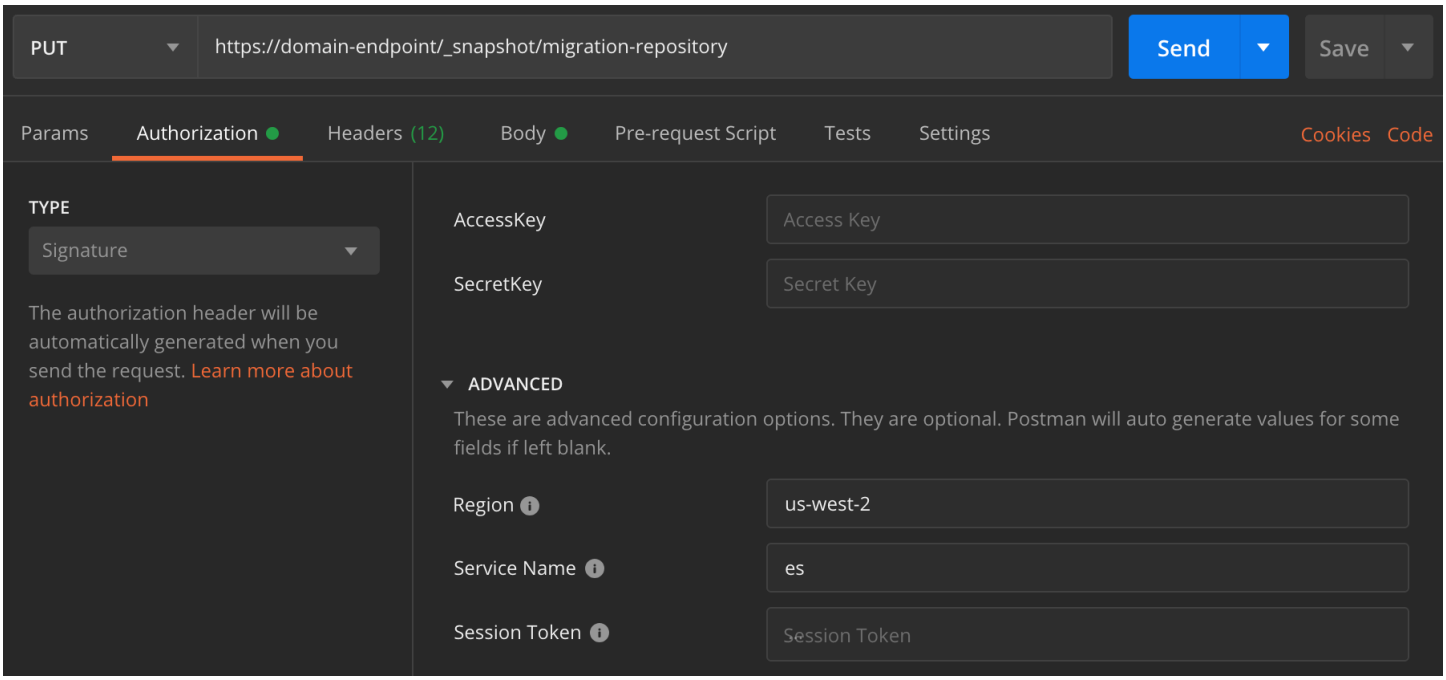
```
arn:aws:iam::123456789123:role/role-name
```

5. PilihPetadan konfirmasi peran muncul di bawahPengguna yang dipetakan.

Memulihkan snapshot

Pada titik ini, Anda memiliki dua cara untuk mengaksesOpenSearchDomain layanan: otentikasi dasar HTTP dengan kredensi pengguna master Anda atauAWSotentikasi menggunakan kredensi IAM Anda. Karena snapshot menggunakan Amazon S3, yang tidak memiliki konsep pengguna utama, Anda harus menggunakan kredensi IAM Anda untuk mendaftarkan repositori snapshot denganOpenSearchDomain layanan.

Sebagian besar bahasa pemrograman memiliki perpustakaan untuk membantu permintaan penandatanganan, tetapi pendekatan yang lebih sederhana adalah dengan menggunakan alat seperti[Tukang Pos](#)dan masukkan mandat IAM Anda ke dalamOtorisasibagian.



Untuk memulihkan snapshot

1. Apa pun cara yang Anda gunakan untuk menandatangani permintaan Anda, langkah pertama adalah mendaftarkan repositori:

```
PUT _snapshot/my-snapshot-repo-name
{
  "type": "s3",
  "settings": {
    "bucket": "bucket-name",
    "region": "us-west-2",
    "role_arn": "arn:aws:iam::123456789012:role/OpenSearchSnapshotRole"
  }
}
```

2. Kemudian daftar snapshot di repositori, dan temukan snapshot yang ingin dipulihkan. Saat ini, Anda dapat terus menggunakan Postman atau beralih ke alat lain seperti curl.

   Singkatan

```
GET _snapshot/my-snapshot-repo-name/_all
```

   meringkuk

```
curl -XGET -u 'master-user:master-user-password' https://domain-endpoint/
_snapshot/my-snapshot-repo-name/_all
```

3. Pulihkan snapshot yang telah disalin.

   Singkatan

```
POST _snapshot/my-snapshot-repo-name/my-snapshot-name/_restore
{
  "indices": "migration-index1,migration-index2,other-indices-*",
  "include_global_state": false
}
```

   meringkuk

```
curl -XPOST -u 'master-user:master-user-password' https://domain-endpoint/
_snapshot/my-snapshot-repo-name/my-snapshot-name/_restore \
```

```
  -H 'Content-Type: application/json' \
  -d '{"indices":"migration-index1,migration-index2,other-indices-
*","include_global_state":false}'
```

4. Akhirnya, verifikasi bahwa indeks Anda dipulihkan seperti yang diharapkan.

   Singkatan

   ```
   GET _cat/indices?v
   ```

   meringkuk

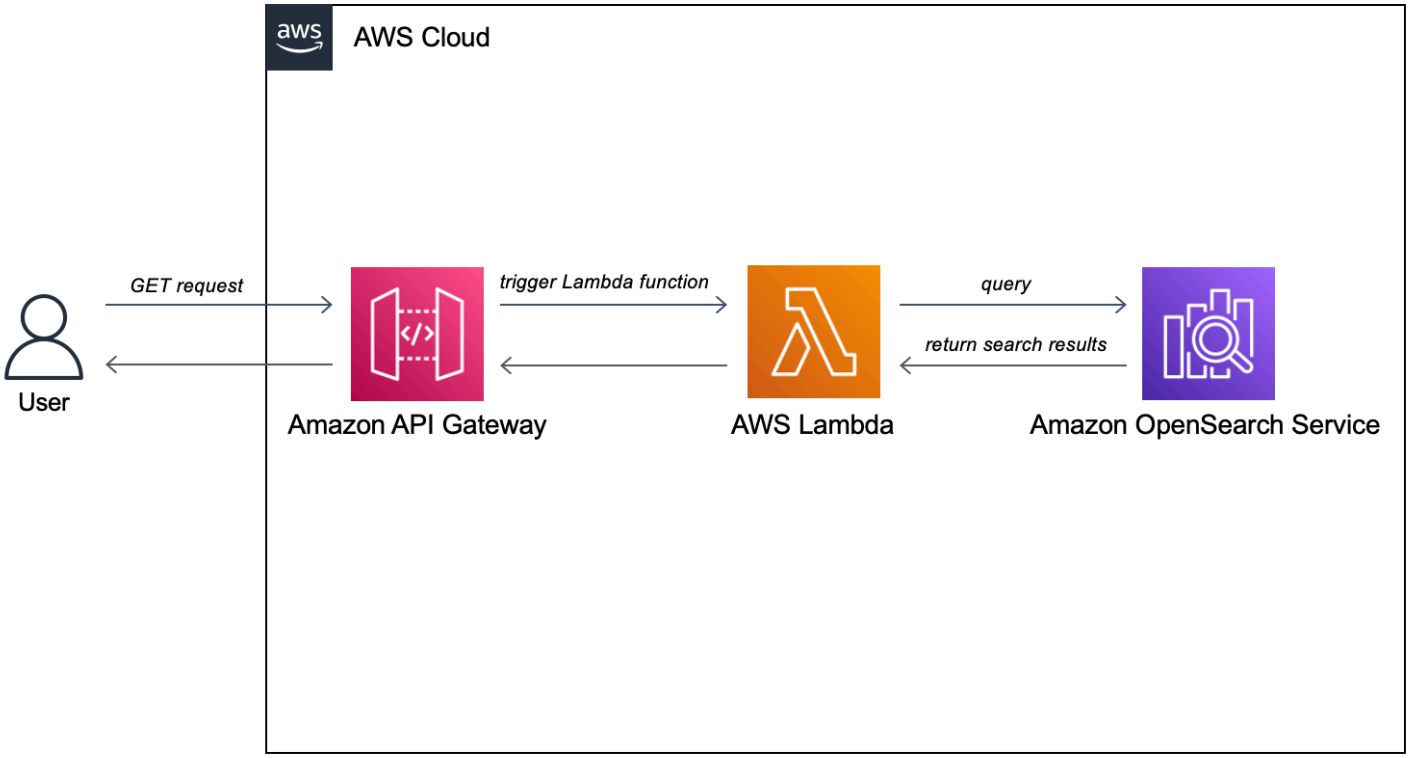
   ```
   curl -XGET -u 'master-user:master-user-password' https://domain-endpoint/_cat/
   indices?v
   ```

Sekarang migrasi telah selesai. Anda dapat mengonfigurasi klien Anda untuk menggunakan yang baruOpenSearchTitik akhir layanan,mengubah ukuran domainagar sesuai dengan beban kerja Anda, periksa jumlah pecahan untuk indeks Anda, beralih kePengguna master IAM, atau mulai membangun visualisasi diOpenSearchDasbor.

Tutorial: Membuat aplikasi pencarian dengan Amazon OpenSearch Service

Cara umum untuk membuat aplikasi pencarian dengan Amazon OpenSearch Service adalah dengan menggunakan formulir web untuk mengirim kueri pengguna ke server. Kemudian Anda dapat mengotorisasi server untuk memanggil OpenSearch API secara langsung dan meminta server mengirim permintaan ke OpenSearch Layanan. Namun, jika Anda ingin menulis kode sisi klien yang tidak bergantung pada server, Anda harus mengkompensasi risiko keamanan dan kinerja. Mengizinkan akses publik yang tidak ditandatangani ke OpenSearch API tidak disarankan. Pengguna mungkin mengakses titik akhir yang tidak aman atau memengaruhi performa klaster melalui kueri yang terlalu luas (atau terlalu banyak kueri).

Bab ini menyajikan solusi: gunakan Amazon API Gateway untuk membatasi pengguna ke subset OpenSearch API dan AWS Lambda untuk menandatangani permintaan dari API Gateway ke OpenSearch Layanan.



> **Note**
>
> Harga Standar API Gateway dan Lambda berlaku, tetapi dalam penggunaan terbatas dari tutorial ini, biaya harus diabaikan.

Prasyarat

Prasyarat untuk tutorial ini adalah domain Layanan. OpenSearch Jika Anda belum memilikinya, ikuti langkah-langkah di Buat domain OpenSearch Layanan untuk membuatnya.

Langkah 1: Mengindeks data sampel

Unduh sample-movies.zip, unzip, lalu gunakan operasi _bulk API untuk menambahkan 5.000 dokumen ke indeks: `movies`

```
POST https://search-my-domain.us-west-1.es.amazonaws.com/_bulk
{ "index": { "_index": "movies", "_id": "tt1979320" } }
{"directors":["Ron
 Howard"],"release_date":"2013-09-02T00:00:00Z","rating":8.3,"genres":
["Action","Biography","Drama","Sport"],"image_url":"http://ia.media-imdb.com/images/
```

```
M/MV5BMTQyMDE0MTY0OV5BMl5BanBnXkFtZTcwMjI2OTI0OQ@@._V1_SX400_.jpg","plot":"A re-
creation of the merciless 1970s rivalry between Formula One rivals James Hunt and
 Niki Lauda.","title":"Rush","rank":2,"running_time_secs":7380,"actors":["Daniel
 Brühl","Chris Hemsworth","Olivia Wilde"],"year":2013,"id":"tt1979320","type":"add"}
{ "index": { "_index": "movies", "_id": "tt1951264" } }
{"directors":["Francis Lawrence"],"release_date":"2013-11-11T00:00:00Z","genres":
["Action","Adventure","Sci-Fi","Thriller"],"image_url":"http://ia.media-imdb.com/
images/M/
MV5BMTAyMjQ3OTAxMzNeQTJeQWpwZ15BbWU4MDU0NzA1MzAx._V1_SX400_.jpg","plot":"Katniss
 Everdeen and Peeta Mellark become targets of the Capitol after
 their victory in the 74th Hunger Games sparks a rebellion in
 the Districts of Panem.","title":"The Hunger Games: Catching
 Fire","rank":4,"running_time_secs":8760,"actors":["Jennifer Lawrence","Josh
 Hutcherson","Liam Hemsworth"],"year":2013,"id":"tt1951264","type":"add"}
...
```

Perhatikan bahwa di atas adalah contoh perintah dengan subset kecil dari data yang tersedia. Untuk melakukan `_bulk` operasi, Anda perlu menyalin dan menempelkan seluruh konten `sample-movies` file. Untuk instruksi lebih lanjut, lihatthe section called “Opsi 2: Unggah beberapa dokumen”.

Anda juga dapat menggunakan perintah curl berikut untuk mencapai hasil yang sama:

```
curl -XPOST -u 'master-user:master-user-password' 'domain-endpoint/_bulk' --data-binary
 @bulk_movies.json -H 'Content-Type: application/json'
```

Langkah 2: Buat dan gunakan fungsi Lambda

Sebelum Anda membuat API di API Gateway, buat fungsi Lambda yang diteruskan permintaannya.

Buat fungsi Lambda

Dalam solusi ini, API Gateway meneruskan permintaan ke fungsi Lambda, yang menanyakan OpenSearch Layanan dan mengembalikan hasil. Karena fungsi sampel ini menggunakan pustaka eksternal, Anda perlu membuat paket penyebaran dan mengunggahnya ke Lambda.

Untuk membuat paket deployment

1. Buka prompt perintah dan membuat direktori proyek `my-opensearch-function`. Misalnya, di macOS:

   ```
   mkdir my-opensearch-function
   ```

2. Arahkan ke direktori proyek `my-sourcecode-function`.

```
cd my-opensearch-function
```

3. Salin isi contoh kode Python berikut dan simpan dalam file baru bernama. `opensearch-lambda.py` Tambahkan Wilayah Anda dan host endpoint ke file.

```
import boto3
import json
import requests
from requests_aws4auth import AWS4Auth

region = '' # For example, us-west-1
service = 'es'
credentials = boto3.Session().get_credentials()
awsauth = AWS4Auth(credentials.access_key, credentials.secret_key, region, service,
 session_token=credentials.token)

host = '' # The OpenSearch domain endpoint with https:// and without a trailing
 slash
index = 'movies'
url = host + '/' + index + '/_search'

# Lambda execution starts here
def lambda_handler(event, context):

    # Put the user query into the query DSL for more accurate search results.
    # Note that certain fields are boosted (^).
    query = {
        "size": 25,
        "query": {
            "multi_match": {
                "query": event['queryStringParameters']['q'],
                "fields": ["title^4", "plot^2", "actors", "directors"]
            }
        }
    }

    # Elasticsearch 6.x requires an explicit Content-Type header
    headers = { "Content-Type": "application/json" }

    # Make the signed HTTP request
    r = requests.get(url, auth=awsauth, headers=headers, data=json.dumps(query))
```

```
    # Create the response and add some extra content to support CORS
    response = {
        "statusCode": 200,
        "headers": {
            "Access-Control-Allow-Origin": '*'
        },
        "isBase64Encoded": False
    }

    # Add the search results to the response
    response['body'] = r.text
    return response
```

4. Instal pustaka eksternal ke `package` direktori baru.

```
pip3 install --target ./package boto3
pip3 install --target ./package requests
pip3 install --target ./package requests_aws4auth
```

5. Buat paket deployment dengan pustaka terinstal di akar. Perintah berikut menghasilkan `my-deployment-package.zip` file di direktori proyek Anda.

```
cd package
zip -r ../my-deployment-package.zip .
```

6. Tambahkan file `opensearch-lambda.py` ke akar dari file zip.

```
cd ..
zip my-deployment-package.zip opensearch-lambda.py
```

Untuk informasi selengkapnya tentang membuat fungsi Lambda dan paket penyebaran, lihat Menerapkan fungsi Lambda Python dengan arsip file.zip di Panduan Pengembang dan dalam panduan ini. AWS Lambda the section called “Membuat paket deployment Lambda”

Untuk membuat fungsi Anda menggunakan konsol Lambda

1. Arahkan ke konsol Lambda di https://console.aws.amazon.com/lambda/home. Di panel navigasi kiri, pilih Fungsi.
2. Pilih Buat fungsi.

3. Konfigurasi bidang berikut:

   - Nama fungsi: opensearch-function
   - Runtime: Python 3.9
   - Arsitektur: x86_64

   Simpan semua opsi default lainnya dan pilih Buat fungsi.
4. Di bagian Sumber kode dari halaman ringkasan fungsi, pilih Unggah dari tarik-turun dan pilih file.zip. Temukan `my-deployment-package.zip` file yang Anda buat dan pilih Simpan.
5. Handler adalah metode dalam fungsi kode Anda yang memproses peristiwa. Di bawah pengaturan Runtime, pilih Edit dan ubah nama handler sesuai dengan nama file dalam paket penyebaran Anda di mana fungsi Lambda berada. Karena file Anda diberi nama`opensearch-lambda.py`, ganti nama handler menjadi. *`opensearch-lambda`*`.lambda_handler` Untuk informasi lebih lanjut, lihat Handler fungsi Lambda di Python.

Langkah 3: Buat API di API Gateway

Menggunakan API Gateway memungkinkan Anda membuat API yang lebih terbatas dan menyederhanakan proses berinteraksi dengan API. OpenSearch `_search` API Gateway memungkinkan Anda mengaktifkan fitur keamanan seperti autentikasi Amazon Cognito dan throttling permintaan. Lakukan langkah-langkah berikut untuk membuat dan men-deploy API:

Buat dan konfigurasi API

Untuk membuat API Anda menggunakan konsol API Gateway

1. Arahkan ke konsol API Gateway di https://console.aws.amazon.com/apigateway/home. Di panel navigasi kiri, pilih API.
2. Cari API REST (tidak privat) dan pilih Membangun.
3. Pada halaman berikut, cari bagian Buat API baru dan pastikan API Baru dipilih.
4. Konfigurasi bidang berikut:

   - Nama API: opensearch-api
   - Deskripsi: API Publik untuk mencari domain OpenSearch Layanan Amazon
   - Tipe Titik Akhir: Regional

5. Pilih Buat API.
6. Pilih Actions dan Create Method.
7. Pilih GET di dropdown dan klik tanda centang untuk mengonfirmasi.
8. Konfigurasikan pengaturan berikut, lalu pilih Simpan:

Pengaturan	Nilai
Tipe integrasi	Fungsi Lambda
Gunakan integrasi proksi Lambda	Ya
Wilayah Lambda	*kami-barat-1*
Fungsi Lambda	`opensearch-lambda`
Gunakan waktu habis default	Ya

Mengkonfigurasi permintaan metode

Pilih Permintaan Metode dan konfigurasikan pengaturan berikut:

Pengaturan	Nilai
Otorisasi	TIDAK ADA
Validator Permintaan	Memvalidasi parameter string kueri dan header
Kunci API Diperlukan	salah

Di bawah Parameter String Kueri URL, pilih Tambahkan string kueri dan konfigurasikan parameter berikut:

Pengaturan	Nilai
Nama	q

Pengaturan	Nilai
Diperlukan	Ya

Men-deploy API dan mengkonfigurasi tahap

Konsol API Gateway memungkinkan Anda men-deploy API dengan membuat deployment dan mengaitkannya dengan tahap baru atau yang sudah ada.

1. Pilih Actions dan Deploy API.
2. Untuk Tahap deployment pilih Tahap Baru dan beri nama tahap `opensearch-api-test`.
3. Pilih Terapkan.
4. Konfigurasikan pengaturan berikut di editor tahap, lalu pilih Simpan Perubahan:

Pengaturan	Nilai
Aktifkan throttling	Ya
Laju	1000
Burst	500

Pengaturan ini mengkonfigurasi API yang hanya memiliki satu metode: permintaan `GET` ke root titik akhir (`https://`*`some-id`*`.execute-api.`*`us-west-1`*`.amazonaws.com/search-es-api-test`). Permintaan membutuhkan parameter tunggal (`q`), string kueri untuk mencari. Ketika dipanggil, metode meneruskan permintaan ke Lambda, yang menjalankan fungsi `opensearch-lambda`. Untuk informasi selengkapnya, lihat Membuat API di Amazon API Gateway dan Menerapkan REST API di Amazon API Gateway.

Langkah 4: (Opsional) Ubah kebijakan akses domain

Domain OpenSearch Layanan Anda harus mengizinkan fungsi Lambda untuk membuat `GET` permintaan ke indeks. `movies` Jika domain Anda memiliki kebijakan akses terbuka dengan kontrol akses halus diaktifkan, Anda dapat membiarkannya apa adanya:

```
{
```

```
  "Version": "2012-10-17",
  "Statement": [
    {
      "Effect": "Allow",
      "Principal": {
        "AWS": "*"
      },
      "Action": "es:*",
      "Resource": "arn:aws:es:us-west-1:123456789012:domain/domain-name/*"
    }
  ]
}
```

Atau, Anda dapat memilih untuk membuat kebijakan akses domain Anda lebih terperinci. Misalnya, kebijakan minimum berikut menyediakan `opensearch-lambda-role` (dibuat melalui Lambda) akses baca ke indeks. `movies` Untuk mendapatkan nama yang tepat dari peran yang dibuat Lambda secara otomatis, pergi ke konsol AWS Identity and Access Management (IAM), pilih Peran, dan cari “lambda”.

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Effect": "Allow",
      "Principal": {
        "AWS": "arn:aws:iam::123456789012:role/service-role/opensearch-lambda-
role-1abcdefg"
      },
      "Action": "es:ESHttpGet",
      "Resource": "arn:aws:es:us-west-1:123456789012:domain/domain-name/movies/_search"
    }
  ]
}
```

> **Important**
>
> Jika Anda mengaktifkan kontrol akses berbutir halus untuk domain, Anda juga perlu memetakan peran tersebut ke pengguna di OpenSearch Dasbor, jika tidak, Anda akan melihat kesalahan izin.

Untuk informasi selengkapnya tentang kebijakan akses, lihat the section called “Mengonfigurasi kebijakan akses”.

Petakan peran Lambda (jika menggunakan kontrol akses berbutir halus)

Kontrol akses berbutir halus memperkenalkan langkah tambahan sebelum Anda dapat menguji aplikasi. Bahkan jika Anda menggunakan otentikasi dasar HTTP untuk semua tujuan lain, Anda perlu memetakan peran Lambda ke pengguna, jika tidak, Anda akan melihat kesalahan izin.

1. Arahkan ke URL OpenSearch Dasbor untuk domain.
2. Dari menu utama, pilih Keamanan, Peran, dan pilih tautan `keall_access`, peran yang Anda butuhkan untuk memetakan peran Lambda.
3. Pilih Pengguna yang Dipetakan, Kelola pemetaan.
4. Di bawah peran Backend, tambahkan Nama Sumber Daya Amazon (ARN) peran Lambda. ARN harus berbentuk. `arn:aws:iam::`*`123456789123`*`:role/service-role/`*`opensearch-lambda-role-1abcdefg`*
5. Pilih Peta dan konfirmasikan pengguna atau peran muncul di bawah Pengguna yang dipetakan.

Langkah 5: Menguji aplikasi web

Untuk menguji aplikasi web

1. Unduh sample-site.zip, unzip itu, dan buka `scripts/search.js` di editor teks favorit Anda.
2. Perbarui `apigatewayendpoint` variabel untuk menunjuk ke titik akhir API Gateway Anda dan tambahkan garis miring terbalik ke akhir jalur yang diberikan. Anda dapat dengan cepat menemukan titik akhir di API Gateway dengan memilih Tahapan dan memilih nama API. `apigatewayendpoint`Variabel harus berbentuk `https://`*`some-id`*`.execute-api.`*`us-west-1`*`.amazonaws.com/opensearch-api-test` /.
3. Buka `index.html` dan coba jalankan pencarian untuk thor, rumah, dan beberapa istilah lainnya.

Movie Search

thor

Found 7 results.

Thor

2011 — The powerful but arrogant god Thor is cast out of Asgard to live amongst humans in Midgard (Earth), where he soon becomes one of their finest defenders.

Thor: The Dark World

2013 — Faced with an enemy that even Odin and Asgard cannot withstand, Thor must embark on his most perilous and personal journey yet, one that will reunite him with Jane Foster and force him to sacrifice everything to save us all.

Vikingdom

2013 — A forgotten king, Eirick, is tasked with the impossible odds to defeat Thor, the God of Thunder.

Memecahkan masalah kesalahan CORS

Meskipun fungsi Lambda menyertakan konten dalam respons untuk mendukung CORS, Anda mungkin masih melihat kesalahan berikut:

```
Access to XMLHttpRequest at '<api-gateway-endpoint>' from origin 'null' has been
 blocked by CORS policy: No 'Access-Control-Allow-Origin' header is present in the
 requested resource.
```

Jika ini terjadi, coba yang berikut ini:

1. Aktifkan CORS pada sumber daya GET. Di bawah Advanced, atur Access-Control-Allow-Credentials ke. `'true'`
2. Menerapkan ulang API Anda di API Gateway (Actions, Deploy API).
3. Hapus dan tambahkan kembali pemicu fungsi Lambda Anda. Tambahkan tambahkan kembali, pilih Tambah pemicu dan buat titik akhir HTTP yang memanggil fungsi Anda. Pemicu harus memiliki konfigurasi berikut:

Pemicu	API	Tahap Deployment	Keamanan
API Gateway	opensearch-api	opensearch-api-test	Buka

Langkah berikutnya

Bab ini hanyalah titik awal untuk mendemonstrasikan sebuah konsep. Anda mungkin mempertimbangkan modifikasi berikut:

- Tambahkan data Anda sendiri ke domain OpenSearch Layanan.
- Menambahkan metode ke API Anda.
- Dalam fungsi Lambda, memodifikasi kueri pencarian atau meningkatkan bidang yang berbeda.
- Gaya hasil secara berbeda atau memodifikasi `search.js` untuk menampilkan bidang yang berbeda kepada pengguna.

Tutorial: Memvisualisasikan panggilan dukungan pelanggan dengan OpenSearch Service dan Dasbor OpenSearch

Bab ini adalah panduan lengkap dari situasi berikut: sebuah bisnis menerima sejumlah panggilan dukungan pelanggan dan ingin menganalisisnya. Apa subjek dari setiap panggilan? Berapa banyak yang positif? Berapa banyak yang negatif? Bagaimana manajer dapat mencari atau meninjau transkrip panggilan ini?

Alur kerja manual mungkin melibatkan karyawan yang mendengarkan rekaman, mencatat subjek dari setiap panggilan, dan memutuskan apakah interaksi pelanggan itu positif atau tidak.

Proses seperti itu akan sangat padat karya. Dengan asumsi waktu rata-rata 10 menit per panggilan, setiap karyawan hanya dapat mendengarkan 48 panggilan per hari. Kecuali bias manusia, data yang mereka hasilkan akan sangat akurat, namun jumlah data akan minimal: hanya subjek panggilan dan sebuah boolean untuk apakah pelanggan puas atau tidak. Apa pun yang lebih terlibat, seperti transkrip lengkap, akan membutuhkan banyak waktu.

Dengan menggunakan Amazon S3, Amazon Transcribe, Amazon Comprehend, dan Amazon OpenSearch Service, Anda dapat mengotomatisasi proses serupa dengan kode yang sangat sedikit dan berakhir dengan lebih banyak data. Misalnya, Anda bisa mendapatkan transkrip lengkap panggilan, kata kunci dari transkrip, dan keseluruhan “sentimen” panggilan (positif, negatif, netral, atau campuran). Kemudian Anda dapat menggunakan OpenSearch dan OpenSearch Dasbor untuk mencari dan memvisualisasikan data.

Meskipun Anda dapat menggunakan panduan ini apa adanya, tujuannya adalah untuk memicu ide-ide tentang bagaimana untuk memperkaya dokumen JSON Anda sebelum Anda mengindeksnya di Service. OpenSearch

Perkiraan Biaya

Secara umum, melakukan langkah-langkah dalam panduan ini membutuhkan biaya kurang dari $2. Panduan menggunakan sumber daya berikut:

- Bucket S3 dengan kurang dari 100 MB yang ditransfer dan disimpan

  Untuk mempelajari selengkapnya, lihat Harga Amazon S3.
- OpenSearchDomain layanan dengan satu `t2.medium` instans dan 10 GIB penyimpanan EBS untuk beberapa jam

  Untuk selengkapnya, lihat Harga Amazon OpenSearch Service.

- Beberapa panggilan ke Amazon Transcribe

  Untuk mempelajari selengkapnya, lihat Harga Amazon Transcribe.
- Beberapa pemrosesan bahasa alami panggilan ke Amazon Comprehend

  Untuk mempelajari selengkapnya, lihat Harga Amazon Comprehend.

Topik

Langkah 1: Mengkonfigurasi prasyarat

Sebelum melanjutkan, Anda harus memiliki sumber daya berikut.

Prasyarat	Deskripsi
Bucket Amazon S3	Untuk informasi selengkapnya, lihat Membuat Bucket di Panduan Pengguna Amazon Simple Storage Service.
OpenSearchDomain layanan	Tujuan untuk data. Untuk selengkapnya, lihat Membuat domain OpenSearch Service.

Jika Anda belum memiliki sumber daya ini, Anda dapat membuatnya menggunakan perintah AWS CLI berikut:

```
aws s3 mb s3://my-transcribe-test --region us-west-2
```

```
aws opensearch create-domain --domain-name my-transcribe-test --engine-version
 OpenSearch_1.0 --cluster-config  InstanceType=t2.medium.search,InstanceCount=1
 --ebs-options EBSEnabled=true,VolumeType=standard,VolumeSize=10 --access-
policies '{"Version":"2012-10-17","Statement":[{"Effect":"Allow","Principal":
```

```
{"AWS":"arn:aws:iam::123456789012:root"},"Action":"es:*","Resource":"arn:aws:es:us-
west-2:123456789012:domain/my-transcribe-test/*"}]}' --region us-west-2
```

> **Note**
>
> Perintah ini menggunakan `us-west-2` Wilayah, tetapi Anda dapat menggunakan Wilayah mana pun yang didukung Amazon Comprehend. Untuk mempelajari lebih lanjut, lihat Referensi Umum AWS.

Langkah 2: Menyalin kode sampel

1. Salin dan tempel kode sampel Python 3 berikut ke dalam file baru bernama `call-center.py`:

```
import boto3
import datetime
import json
import requests
from requests_aws4auth import AWS4Auth
import time
import urllib.request

# Variables to update
audio_file_name = '' # For example, 000001.mp3
bucket_name = '' # For example, my-transcribe-test
domain = '' # For example, https://search-my-transcribe-test-12345.us-
west-2.es.amazonaws.com
index = 'support-calls'
type = '_doc'
region = 'us-west-2'

# Upload audio file to S3.
s3_client = boto3.client('s3')

audio_file = open(audio_file_name, 'rb')

print('Uploading ' + audio_file_name + '...')
response = s3_client.put_object(
    Body=audio_file,
    Bucket=bucket_name,
    Key=audio_file_name
)
```

```
# # Build the URL to the audio file on S3.
# # Only for the us-east-1 region.
# mp3_uri = 'https://' + bucket_name + '.s3.amazonaws.com/' + audio_file_name

# Get the necessary details and build the URL to the audio file on S3.
# For all other regions.
response = s3_client.get_bucket_location(
    Bucket=bucket_name
)
bucket_region = response['LocationConstraint']
mp3_uri = 'https://' + bucket_name + '.s3-' + bucket_region + '.amazonaws.com/' +
 audio_file_name

# Start transcription job.
transcribe_client = boto3.client('transcribe')

print('Starting transcription job...')
response = transcribe_client.start_transcription_job(
    TranscriptionJobName=audio_file_name,
    LanguageCode='en-US',
    MediaFormat='mp3',
    Media={
        'MediaFileUri': mp3_uri
    },
    Settings={
        'ShowSpeakerLabels': True,
        'MaxSpeakerLabels': 2 # assumes two people on a phone call
    }
)

# Wait for the transcription job to finish.
print('Waiting for job to complete...')
while True:
    response =
 transcribe_client.get_transcription_job(TranscriptionJobName=audio_file_name)
    if response['TranscriptionJob']['TranscriptionJobStatus'] in ['COMPLETED',
 'FAILED']:
        break
    else:
        print('Still waiting...')
    time.sleep(10)

transcript_uri = response['TranscriptionJob']['Transcript']['TranscriptFileUri']
```

```
# Open the JSON file, read it, and get the transcript.
response = urllib.request.urlopen(transcript_uri)
raw_json = response.read()
loaded_json = json.loads(raw_json)
transcript = loaded_json['results']['transcripts'][0]['transcript']

# Send transcript to Comprehend for key phrases and sentiment.
comprehend_client = boto3.client('comprehend')

# If necessary, trim the transcript.
# If the transcript is more than 5 KB, the Comprehend calls fail.
if len(transcript) > 5000:
    trimmed_transcript = transcript[:5000]
else:
    trimmed_transcript = transcript

print('Detecting key phrases...')
response = comprehend_client.detect_key_phrases(
    Text=trimmed_transcript,
    LanguageCode='en'
)

keywords = []
for keyword in response['KeyPhrases']:
    keywords.append(keyword['Text'])

print('Detecting sentiment...')
response = comprehend_client.detect_sentiment(
    Text=trimmed_transcript,
    LanguageCode='en'
)

sentiment = response['Sentiment']

# Build the Amazon OpenSearch Service URL.
id = audio_file_name.strip('.mp3')
url = domain + '/' + index + '/' + type + '/' + id

# Create the JSON document.
json_document = {'transcript': transcript, 'keywords': keywords, 'sentiment':
 sentiment, 'timestamp': datetime.datetime.now().isoformat()}

# Provide all details necessary to sign the indexing request.
```

```
credentials = boto3.Session().get_credentials()
awsauth = AWS4Auth(credentials.access_key, credentials.secret_key, region,
 'opensearchservice', session_token=credentials.token)

# Index the document.
print('Indexing document...')
response = requests.put(url, auth=awsauth, json=json_document, headers=headers)

print(response)
print(response.json())
```

2. Perbarui enam variabel awal.
3. Instal paket yang diperlukan menggunakan perintah-perintah berikut:

```
pip install boto3
pip install requests
pip install requests_aws4auth
```

4. Tempatkan MP3 Anda di direktori yang sama dengan `call-center.py` dan jalankan skrip. Sebuah output sampel berikut:

```
$ python call-center.py
Uploading 000001.mp3...
Starting transcription job...
Waiting for job to complete...
Still waiting...
Still waiting...
Still waiting...
Still waiting...
Still waiting...
Still waiting...
Still waiting...
Detecting key phrases...
Detecting sentiment...
Indexing document...
<Response [201]>
{u'_type': u'call', u'_seq_no': 0, u'_shards': {u'successful': 1, u'failed': 0,
 u'total': 2}, u'_index': u'support-calls4', u'_version': 1, u'_primary_term': 1,
 u'result': u'created', u'_id': u'000001'}
```

`call-center.py` melakukan sejumlah operasi:

1. Skrip mengunggah file audio (dalam hal ini, MP3, namun Amazon Transcribe mendukung beberapa format) ke bucket S3 Anda.
2. Ini mengirimkan URL file audio ke Amazon Transcribe dan menunggu tugas transkripsi selesai.

   Waktu untuk menyelesaikan tugas transkripsi tergantung pada panjang file audio. Asumsikan menit, bukan detik.

   > **Tip**
   >
   > Untuk meningkatkan kualitas transkripsi, Anda dapat mengkonfigurasi kosa kata kustom untuk Amazon Transcribe.

3. Setelah tugas transkripsi selesai, skrip mengekstraksi transkrip, memangkasnya menjadi 5.000 karakter, dan mengirimkannya ke Amazon Comprehend untuk analisis kata kunci dan sentimen.
4. Terakhir, skrip menambahkan transkrip lengkap, kata kunci, sentimen, stempel waktu saat ini ke dokumen JSON dan mengindeksnya di Service. OpenSearch

> **Tip**
>
> LibriVoxmemiliki buku audio domain publik yang dapat Anda gunakan untuk pengujian.

(Opsional) Langkah 3: Mengindeks data sampel

Jika Anda tidak memiliki banyak rekaman panggilan berguna—dan siapa yang melakukannya?—Anda dapat mengindeks dokumen sampel di sample-calls.zip, yang sebanding dengan apa yang dihasilkan `call-center.py`.

1. Buat file bernama `bulk-helper.py`:

```
import boto3
from opensearchpy import OpenSearch, RequestsHttpConnection
import json
from requests_aws4auth import AWS4Auth

host = '' # For example, my-test-domain.us-west-2.es.amazonaws.com
region = '' # For example, us-west-2
service = 'es'
```

```
bulk_file = open('sample-calls.bulk', 'r').read()

credentials = boto3.Session().get_credentials()
awsauth = AWS4Auth(credentials.access_key, credentials.secret_key, region, service,
 session_token=credentials.token)

search = OpenSearch(
    hosts = [{'host': host, 'port': 443}],
    http_auth = awsauth,
    use_ssl = True,
    verify_certs = True,
    connection_class = RequestsHttpConnection
)

response = search.bulk(bulk_file)
print(json.dumps(response, indent=2, sort_keys=True))
```

2. Perbarui dua variabel awal untuk `host` dan `region`.
3. Instal paket yang diperlukan menggunakan perintah berikut:

```
pip install opensearch-py
```

4. Unduh dan unzip sample-calls.zip.
5. Tempatkan `sample-calls.bulk` dalam direktori yang sama dengan `bulk-helper.py` dan jalankan pembantu. Sebuah output sampel berikut:

```
$ python bulk-helper.py
{
  "errors": false,
  "items": [
    {
      "index": {
        "_id": "1",
        "_index": "support-calls",
        "_primary_term": 1,
        "_seq_no": 42,
        "_shards": {
          "failed": 0,
          "successful": 1,
          "total": 2
        },
        "_type": "_doc",
```

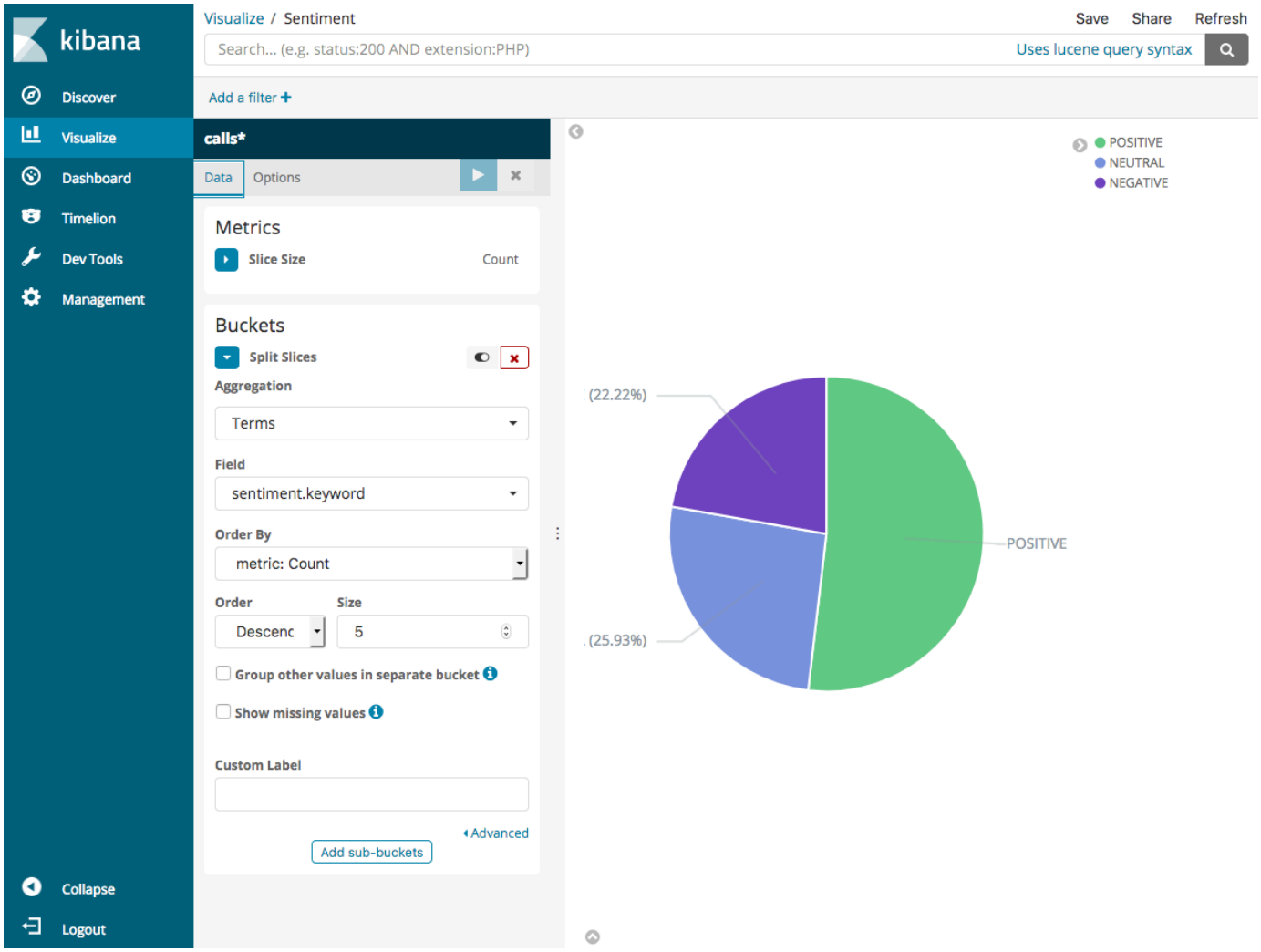
```
        "_version": 9,
        "result": "updated",
        "status": 200
      }
    },
    ...
  ],
  "took": 27
}
```

Langkah 4: Menganalisis dan memvisualisasikan data Anda

Sekarang setelah Anda memiliki beberapa data di OpenSearch Service, Anda dapat memvisualisasikannya menggunakan OpenSearch Dasbor.

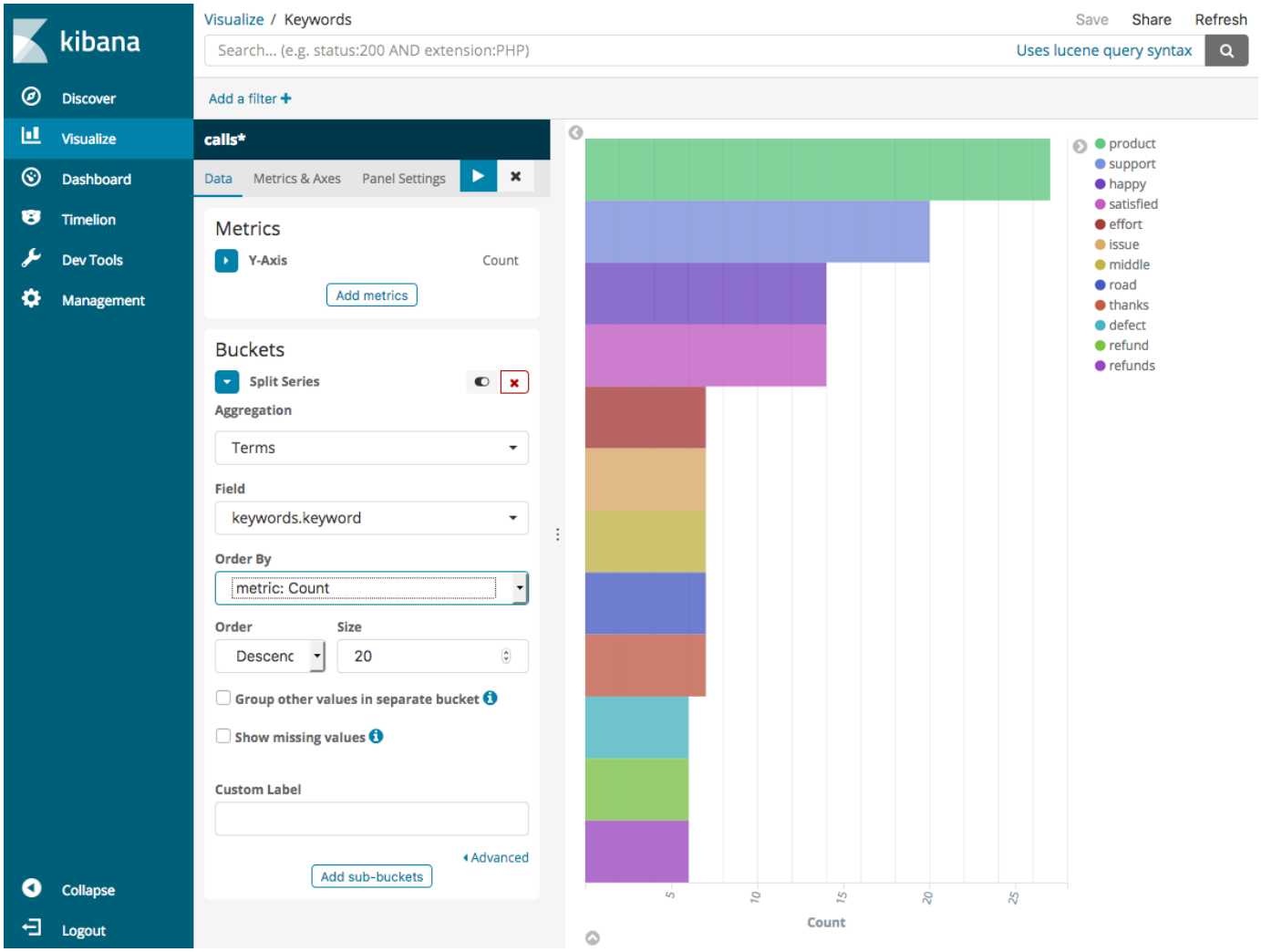
1. Navigasikan ke `https://search-`*`domain`*`.`*`region`*`.es.amazonaws.com/_dashboards`.
2. Sebelum Anda dapat menggunakan OpenSearch Dasbor, Anda memerlukan pola indeks. Dasbor menggunakan pola indeks untuk mempersempit analisis Anda menjadi satu atau lebih indeks. Untuk mencocokkan `support-calls` indeks yang `call-center.py` dibuat, pergi ke Stack Management, Index Patterns, dan menentukan pola indeks`support*`, dan kemudian pilih Next step.
3. Untuk Nama bidang Filter Waktu, pilih stempel waktu.
4. Sekarang Anda dapat mulai membuat visualisasi. Pilih Visualisasikan, lalu tambahkan visualisasi baru.
5. Pilih diagram lingkaran dan pola indeks `support*`.
6. Visualisasi default adalah dasar, jadi pilih Belah Irisan untuk membuat visualisasi yang lebih menarik.

   Untuk Agregasi, pilih Persyaratan. Untuk Bidang, pilih sentiment.keyword. Lalu pilih Terapkan perubahan dan Simpan.

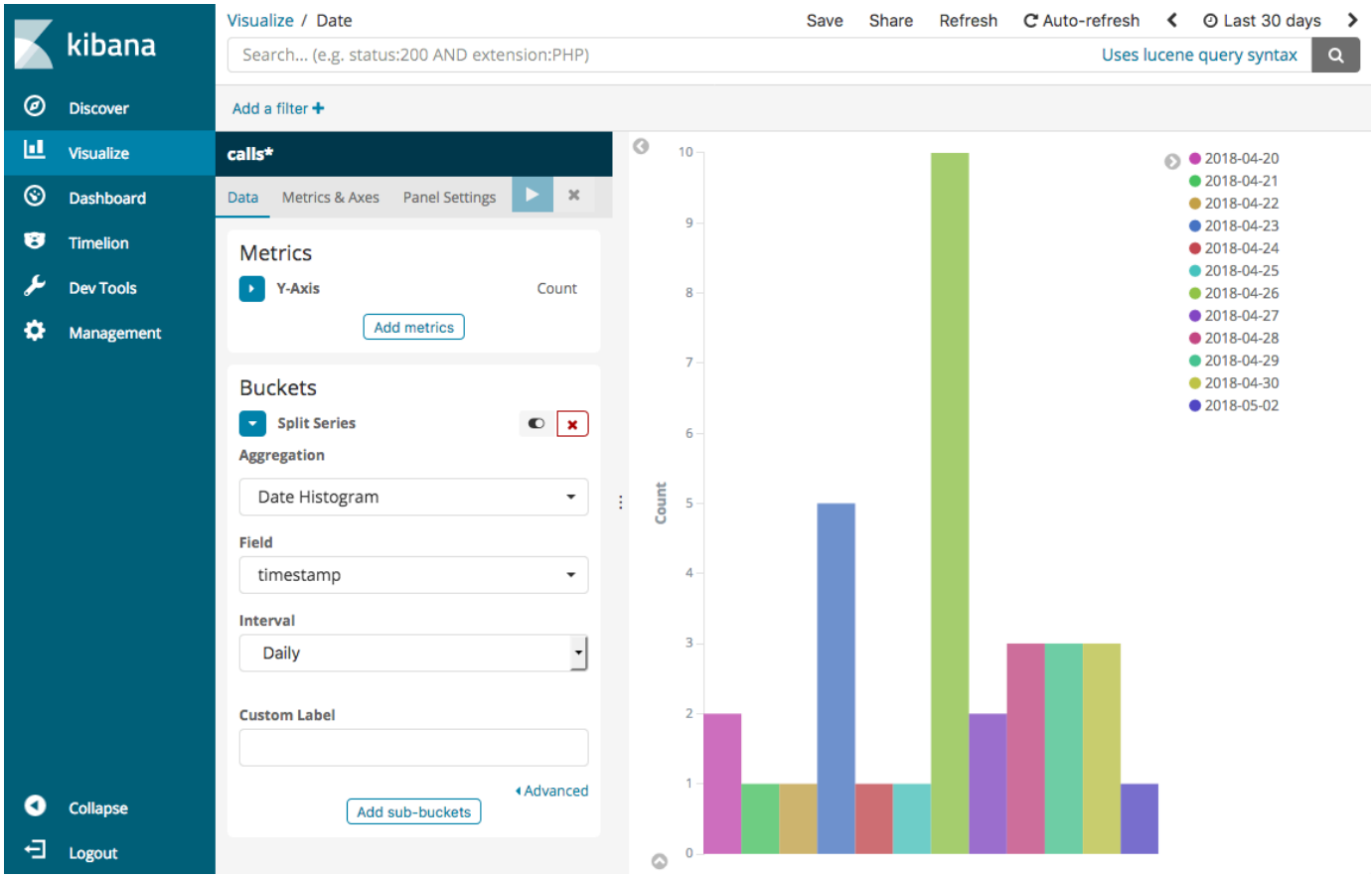


7. Kembali ke halaman Visualisasikan, dan tambahkan visualisasi lain. Kali ini, pilih diagram batang horizontal.
8. Pilih Bagi Seri.

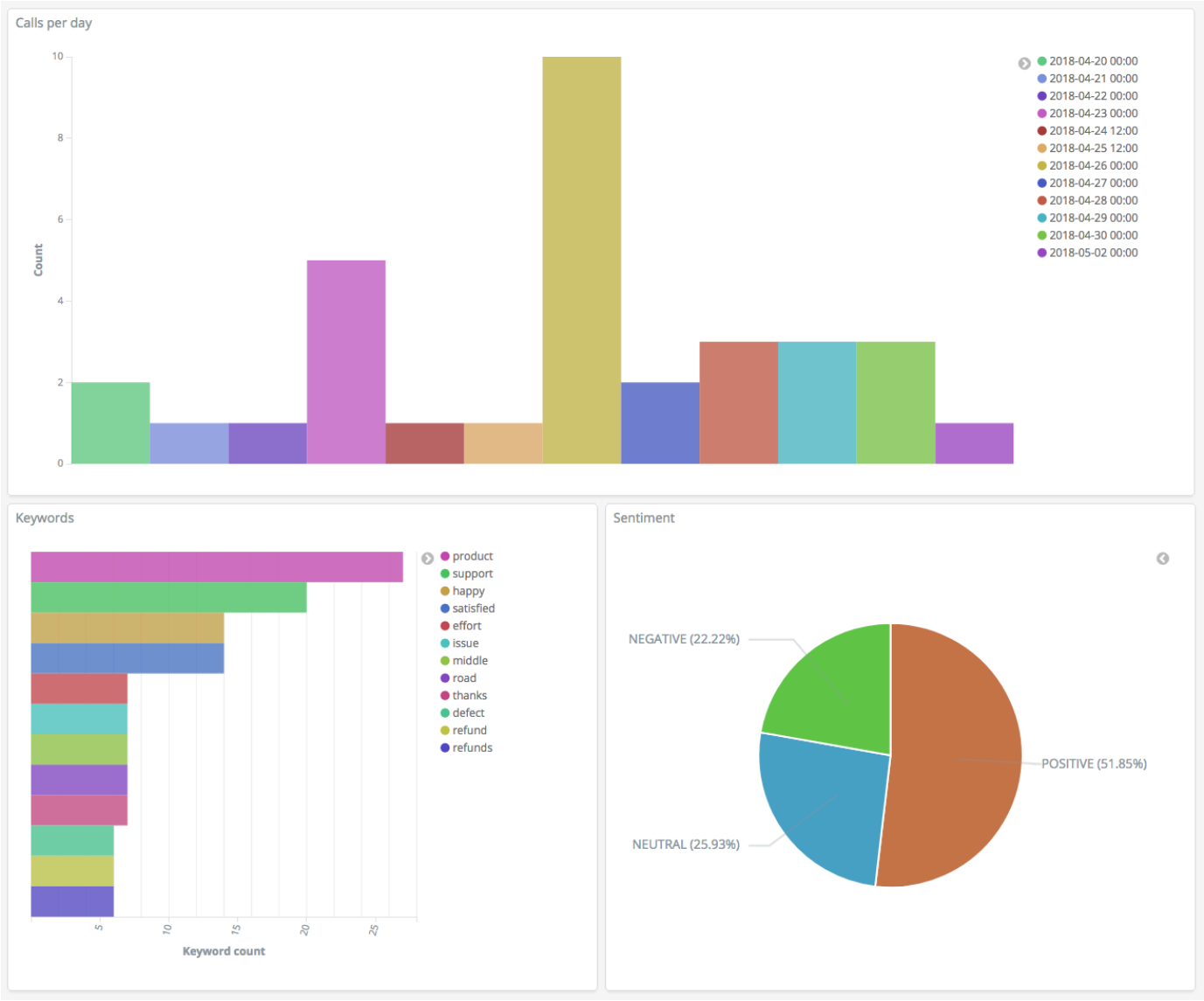
   Untuk Agregasi, pilih Persyaratan. Untuk Bidang, pilih keywords.keyword dan ubah Ukuran ke 20. Lalu pilih Terapkan Perubahan dan Simpan.



9. Kembali ke halaman Visualisasikan dan tambahkan satu visualisasi terakhir, diagram batang vertikal.
10. Pilih Bagi Seri. Untuk Agregasi, pilih Histogram Tanggal. Untuk Bidang, pilih stempel waktu dan ubah Interval ke Harian.
11. Pilih Metrik & Sumbu dan ubah Mode ke Normal.
12. Pilih Terapkan Perubahan dan Simpan.



13. Sekarang setelah Anda memiliki tiga visualisasi, Anda dapat menambahkannya ke visualisasi dasbor. Pilih Dasbor, buat sebuah dasbor, dan tambahkan visualisasi Anda.



Langkah 5: Membersihkan sumber daya dan langkah selanjutnya

Untuk menghindari biaya yang tidak perlu, hapus bucket S3 dan domain OpenSearch Service. Untuk mempelajari selengkapnya, lihat Menghapus Bucket di Panduan Pengguna Amazon Simple Storage OpenSearch Service dan Hapus domain Layanan dalam panduan ini.

Transkrip memerlukan ruang disk yang jauh lebih sedikit daripada file MP3. Anda mungkin dapat mempersingkat jendela penyimpanan MP3 Anda—misalnya, dari tiga bulan rekaman panggilan menjadi satu bulan—mempertahankan transkrip selama bertahun-tahun, dan tetap menghemat biaya penyimpanan.

Anda juga dapat mengotomatisasi proses transkripsi menggunakan AWS Step Functions dan Lambda, tambahkan metadata tambahan sebelum pengindeksan, atau buat visualisasi yang lebih kompleks agar sesuai dengan kasus penggunaan Anda yang tepat.

Amazon OpenSearch Service rename - Ringkasan perubahan

Pada 8 September 2021, rangkaian pencarian dan analitik kami diubah namanya menjadi Amazon OpenSearch Service. OpenSearch Layanan mendukung OpenSearch serta warisan Elasticsearch OSS. Bagian berikut menjelaskan berbagai bagian layanan yang berubah dengan penggantian nama, dan tindakan apa yang perlu Anda lakukan untuk memastikan bahwa domain Anda terus berfungsi dengan baik.

Beberapa perubahan ini hanya berlaku saat Anda meningkatkan domain dari Elasticsearch ke OpenSearch. Dalam kasus lain, seperti di konsol Billing and Cost Management, pengalaman akan segera berubah.

Perhatikan bahwa daftar ini bukan daftar lengkap. Sementara bagian lain dari produk juga berubah, pembaruan ini adalah yang paling relevan.

Topik

Versi API baru

Versi baru API konfigurasi OpenSearch Layanan (2021-01-01) berfungsi dengan OpenSearch serta Elasticsearch OSS lama. 21 operasi API diganti dengan nama yang lebih ringkas dan engine-agnostik (misalnya,`CreateElasticsearchDomain` diubah menjadi`CreateDomain`), tetapi OpenSearch Service terus mendukung kedua versi API.

Kami menyarankan agar Anda menggunakan operasi API baru untuk membuat dan mengelola domain ke depan. Perhatikan bahwa ketika Anda menggunakan operasi API baru untuk membuat domain, Anda perlu menentukan`EngineVersion` parameter dalam format`Elasticsearch_X.Y` atau`OpenSearch_X.Y`, bukan hanya nomor versi. Jika Anda tidak menentukan versinya, itu akan mengatur default ke versi terbaru dari OpenSearch.

Upgrade AndaAWS CLI ke versi 1.20.40 atau yang lebih baru`aws opensearch ...` untuk digunakan untuk membuat dan mengelola domain Anda. Untuk format CLI baru, lihat referensiOpenSearch CLI.

Tipe instans berganti nama

Jenis instans di Amazon OpenSearch Service sekarang dalam format`<type>.<size>.search` — misalnya,`m6g.large.`*`search`* bukan`m6g.large.`*`elasticsearch`*. Anda tidak perlu mengambil tindakan apa pun. Domain yang ada akan mulai secara otomatis merujuk ke jenis instans baru di dalam API dan di konsol Billing and Cost Management.

Jika Anda memiliki Instans Cadangan (RI), kontrak Anda tidak akan terpengaruh oleh perubahan tersebut. Versi API konfigurasi lama masih kompatibel dengan format penamaan lama, tetapi jika Anda ingin menggunakan versi API baru, Anda perlu menggunakan format baru.

Perubahan kebijakan akses

Bagian berikut menjelaskan tindakan apa yang perlu Anda lakukan untuk memperbarui kebijakan akses Anda.

Kebijakan IAM

Kami menyarankan agar Anda memperbarui kebijakan IAM untuk menggunakan operasi API berganti nama. Namun, OpenSearch Layanan akan terus menghormati kebijakan yang ada dengan mereplikasi izin API lama secara internal. Misalnya, jika saat ini Anda memiliki izin untuk melakukan`CreateElasticsearchDomain` operasi, Anda sekarang dapat melakukan panggilan ke keduanya`CreateElasticsearchDomain` (operasi API lama) dan`CreateDomain` (operasi API baru). Hal yang sama berlaku untuk penyangkalan eksplisit. Untuk melihat daftar operasi API yang diperbarui, lihat referensi elemen kebijakan.

Kebijakan SCP

Kebijakan kontrol layanan (SCP) memperkenalkan lapisan kompleksitas tambahan dibandingkan dengan IAM standar. Untuk mencegah kebijakan SCP Anda melanggar, Anda perlu menambahkan operasi API lama dan baru ke setiap kebijakan SCP Anda. Misalnya, jika pengguna saat ini mengizinkan izin`CreateElasticsearchDomain`, Anda juga perlu memberi mereka izin izin`CreateDomain` agar mereka dapat mempertahankan kemampuan untuk membuat domain. Hal yang sama berlaku untuk penyangkalan eksplisit.

Misalnya:

```
"Statement": [
    {
      "Effect": "Allow",
      "Action": [
        "es:CreateElasticsearchDomain",
        "es:CreateDomain"
         ...
      ],
    },
      "Effect": "Deny",
      "Action:" [
        "es:DeleteElasticsearchDomain",
        "es:DeleteDomain"
         ...
```

Tipe sumber daya baru

OpenSearch Layanan memperkenalkan jenis sumber daya baru berikut:

Sumber Daya	Deskripsi
`AWS::OpenSearchService::Domain`	Merupakan domain OpenSearch Layanan Amazon. Sumber daya ini ada di tingkat layanan dan tidak spesifik untuk perangkat lunak yang berjalan di domain. Ini berlaku untuk layanan seperti AWS CloudFormationdan AWSResource Groups, di mana Anda

Sumber Daya	Deskripsi
	membuat dan mengelola sumber daya untuk layanan secara keseluruhan. Untuk petunjuk untuk meningkatkan domain yang didefinisikan dalam CloudFormation dari Elasticsearch ke OpenSearch, lihat Keterangan di Panduan CloudFormation Pengguna.
`AWS::OpenSearch::Domain`	OpenSearchMewakili/Elasticsearch perangkat lunak yang berjalan pada domain. Sumber daya ini berlaku untuk layanan seperti AWS CloudTraildan AWS Config, yang mereferen sikan perangkat lunak yang berjalan pada domain, bukan OpenSearch Layanan secara keseluruhan. Layanan ini sekarang berisi jenis sumber daya terpisah untuk domain yang menjalankan Elasticsearch (`AWS::Elasticsearch::Domain`) versus domain running OpenSearch (`AWS::OpenSearch::Domain`).

> **Note**
>
> Di AWS Config, Anda akan terus melihat data Anda di bawah jenis`AWS::Elasticsearch::Domain` sumber daya yang ada selama beberapa minggu, bahkan jika Anda meningkatkan satu atau beberapa domain ke OpenSearch.

Kibana berganti nama menjadi OpenSearch Dasbor

OpenSearch Dasbor,AWS alternatif untuk Kibana, adalah alat visualisasi open-source yang dirancang untuk digunakan OpenSearch. Setelah Anda memutakhirkan domain dari Elasticsearch ke OpenSearch,`/_plugin/kibana` titik akhir akan berubah menjadi`/_dashboards`. OpenSearch Layanan akan mengalihkan semua permintaan ke endpoint baru, tetapi jika Anda menggunakan

endpoint Kibana dalam salah satu kebijakan IAM Anda, perbarui kebijakan tersebut untuk menyertakan/_dashboards endpoint baru juga.

Jika Anda menggunakanthe section called “Otentikasi SAMP untuk Dasbor OpenSearch ”, sebelum meningkatkan domain OpenSearch, Anda perlu mengubah semua URL Kibana yang dikonfigurasi di penyedia identitas (IdP) dari/_plugin/kibana ke/_dashboards. URL yang paling umum adalah URL assertion consumer service (ACS) dan URL penerima.

kibana_read_onlyPeran default untuk OpenSearch Dasbor diubah namanya menjadiopensearch_dashboards_read_only, dankibana_user peran diubah namanyaopensearch_dashboards_user. Perubahan ini berlaku untuk semua yang baru dibuat OpenSearch 1. x domain yang menjalankan perangkat lunak layanan R20211203 atau yang lebih baru. Jika Anda meningkatkan domain yang ada ke perangkat lunak layanan R20211203, nama peran tetap sama.

CloudWatch Metrik berganti nama

Beberapa CloudWatch metrik berubah untuk domain yang berjalan OpenSearch. Saat Anda meningkatkan domain ke OpenSearch, metrik berubah secara otomatis dan CloudWatch alarm Anda saat ini akan rusak. Sebelum memutakhirkan klaster Anda dari versi Elasticsearch ke OpenSearch versi, pastikan untuk memperbarui CloudWatch alarm Anda untuk menggunakan metrik baru.

Metrik berikut berubah:

Nama metrik asli	Nama baru
KibanaHealthyNodes	OpenSearchDashboardsHealthyNodes
KibanaConcurrentConnections	OpenSearchDashboardsConcurrentConnections
KibanaHeapTotal	OpenSearchDashboardsHeapTotal
KibanaHeapUsed	OpenSearchDashboardsHeapUsed
KibanaHeapUtilization	OpenSearchDashboardsHeapUtilization

Nama metrik asli	Nama baru
KibanaOS1MinuteLoad	OpenSearchDashboardsOS1MinuteLoad
KibanaRequestTotal	OpenSearchDashboardsRequestTotal
KibanaResponseTimesMaxInMillis	OpenSearchDashboardsResponseTimesMaxInMillis
ESReportingFailedRequestSysErrCount	KibanaReportingFailedRequestSysErrCount
ESReportingRequestCount	KibanaReportingRequestCount
ESReportingFailedRequestUserErrCount	KibanaReportingFailedRequestUserErrCount
ESReportingSuccessCount	KibanaReportingSuccessCount
ElasticsearchRequests	OpenSearchRequests

Untuk daftar lengkap metrik yang dikirim OpenSearch Layanan ke Amazon CloudWatch, lihatthe section called “Memantau metrik klaster”.

Perubahan konsol Billing and Cost Management

Data historis di konsol Penagihan dan Manajemen Biaya dan dalam Laporan Biaya dan Penggunaan akan terus menggunakan nama layanan lama, jadi Anda harus mulai menggunakan filter untuk Amazon OpenSearch Service dan nama Elasticsearch lama saat mencari data. Jika Anda memiliki laporan tersimpan yang ada, perbarui filter untuk memastikannya juga menyertakan OpenSearch Layanan. Anda mungkin awalnya menerima peringatan ketika penggunaan Anda berkurang untuk Elasticsearch dan meningkat untuk OpenSearch, tetapi menghilang dalam beberapa hari.

Selain nama layanan, kolom berikut akan berubah untuk semua laporan, tagihan, dan operasi API daftar harga:

Bidang	Format lama	Format baru
Tipe instans	`m5.large.elasticsearch`	`m5.large.search`
Keluarga produk	Contoh Elasticsearch Volume Elasticsearch	Instans OpenSearch Layanan Amazon Volume OpenSearch Layanan Amazon
Deskripsi harga	$5,098 per c5.18xlarge.elasticsearch instans hour (atau sebagian jam) - UE	$5,098 per c5.18xlarge.search instans hour (atau sebagian jam) - EU
Keluarga instance	`ultrawarm.elasticsearch`	`ultrawarm.search`

Format peristiwa baru

Format peristiwa yang dikirim OpenSearch Layanan ke Amazon EventBridge dan Amazon CloudWatch telah berubah, khususnya`detail-type` bidangnya. Bidang sumber (`aws.es`) tetap sama. Untuk format lengkap untuk setiap jenis acara, lihatthe section called “Pemantauan peristiwa”. Jika Anda memiliki aturan acara yang ada yang bergantung pada format lama, pastikan untuk memperbaruinya agar sesuai dengan format baru.

Apa yang tetap sama?

Fitur dan fungsionalitas berikut, antara lain yang tidak terdaftar, akan tetap sama:

- Prinsipal layanan (`es.amazonaws.com`)
- Kode Vendor
- ARN Domain
- Titik akhir domain

Memulai: Tingkatkan domain Anda ke OpenSearch 1.x

OpenSearch 1. x mendukung upgrade dari Elasticsearch versi 6.8 dan 7. x. Untuk petunjuk untuk meningkatkan domain Anda, lihatthe section called “Memulai upgrade (konsol)”. Jika Anda menggunakanAWS CLI atau API konfigurasi untuk meningkatkan domain Anda, Anda perlu menentukan`TargetVersion` sebagai`OpenSearch_1.x`.

OpenSearch 1. x memperkenalkan setelan domain tambahan yang disebut Aktifkan mode kompatibilitas. Karena klien dan plugin Elasticsearch OSS tertentu memeriksa versi klaster sebelum menghubungkan, mode OpenSearch kompatibilitas akan melaporkan versinya sebagai 7.10 sehingga klien ini terus bekerja.

Anda dapat mengaktifkan mode kompatibilitas saat membuat OpenSearch domain untuk pertama kalinya, atau saat Anda memutakhirkan OpenSearch ke versi Elasticsearch. Jika tidak disetel, parameter akan ditetapkan secara default`false` saat Anda membuat domain, dan`true` saat Anda meningkatkan domain.

Untuk mengaktifkan mode kompatibilitas menggunakan API konfigurasi, atur`override_main_response_version` ke`true`:

```
POST https://es.us-east-1.amazonaws.com/2021-01-01/opensearch/upgradeDomain
{
  "DomainName": "domain-name",
  "TargetVersion": "OpenSearch_1.0",
  "AdvancedOptions": {
    "override_main_response_version": "true"
   }
}
```

Untuk mengaktifkan atau menonaktifkan mode kompatibilitas pada OpenSearch domain yang ada, Anda perlu menggunakan operasi API OpenSearch _cluster/settings:

```
PUT /_cluster/settings
{
  "persistent" : {
    "compatibility.override_main_response_version" : true
  }
}
```

Memecahkan Masalah Layanan Amazon OpenSearch

Topik ini menjelaskan cara mengidentifikasi dan memecahkan masalah OpenSearch Layanan Amazon yang umum. Konsultasikan informasi di bagian ini sebelum menghubungi AWS Support.

Tidak dapat mengakses OpenSearch Dasbor

Titik akhir OpenSearch Dasbor tidak mendukung permintaan yang ditandatangani. Jika kebijakan kontrol akses untuk domain Anda hanya memberikan akses ke peran IAM tertentu dan Anda belum mengonfigurasi autentikasi Amazon Cognito, Anda mungkin menerima kesalahan berikut saat mencoba mengakses Dasbor:

```
"User: anonymous is not authorized to perform: es:ESHttpGet"
```

Jika domain OpenSearch Layanan Anda menggunakan akses VPC, Anda mungkin tidak menerima kesalahan ini, tetapi permintaan tersebut mungkin habis. Untuk mempelajari selengkapnya tentang memperbaiki masalah ini dan berbagai opsi konfigurasi yang tersedia untuk Anda, lihat the section called “Mengontrol akses ke OpenSearch Dasbor”, the section called “Tentang kebijakan akses pada domain VPC”, dan the section called “Manajemen Identitas dan Akses”.

Tidak dapat mengakses domain VPC

Lihat the section called “Tentang kebijakan akses pada domain VPC” dan the section called “Menguji domain VPC”.

Klaster dalam status hanya-baca

Dibandingkan dengan versi Elasticsearch sebelumnya, OpenSearch dan Elasticsearch 7. x menggunakan sistem yang berbeda untuk koordinasi cluster. Dalam sistem baru ini, ketika klaster kehilangan kuorum, klaster ini tidak tersedia sampai Anda mengambil tindakan. Kehilangan kuorum dapat terjadi dalam dua bentuk:

- Jika klaster Anda menggunakan simpul utama terdedikasi, kehilangan kuorum terjadi ketika setengah atau lebih dari klaster tidak tersedia.
- Jika klaster Anda tidak menggunakan simpul utama terdedikasi, kehilangan kuorum terjadi ketika setengah atau lebih dari simpul data Anda tidak tersedia.

Jika kehilangan kuorum terjadi dan klaster Anda memiliki lebih dari satu node, OpenSearch Service mengembalikan kuorum dan menempatkan cluster ke status hanya-baca. Anda memiliki dua opsi:

- Menghapus status hanya-baca dan menggunakan klaster apa adanya.
- Kembalikan indeks cluster atau individu dari snapshot.

Jika Anda lebih suka menggunakan klaster apa adanya, verifikasikan bahwa kesehatan klaster hijau menggunakan permintaan berikut:

```
GET _cat/health?v
```

Jika kesehatan klaster merah, kami sarankan memulihkan klaster dari snapshot. Anda juga dapat melihat the section called “Status klaster merah” untuk langkah-langkah pemecahan masalah. Jika kesehatan klaster berwarna hijau, periksa apakah semua indeks yang diharapkan hadir menggunakan permintaan berikut:

```
GET _cat/indices?v
```

Kemudian jalankan beberapa pencarian untuk memverifikasi bahwa data yang diharapkan ada. Jika ada, Anda dapat menghapus status hanya-baca menggunakan permintaan berikut:

```
PUT _cluster/settings
{
  "persistent": {
    "cluster.blocks.read_only": false
  }
}
```

Jika kehilangan kuorum terjadi dan klaster Anda hanya memiliki satu node, OpenSearch Service menggantikan node dan tidak menempatkan cluster ke status read-only. Jika tidak, opsi Anda sama: menggunakan klaster apa adanya atau memulihkan dari snapshot.

Dalam kedua situasi tersebut, OpenSearch Layanan mengirimkan dua acara ke Anda AWS Health Dashboard. Yang pertama memberitahu Anda tentang hilangnya kuorum. Yang kedua terjadi setelah OpenSearch Layanan berhasil mengembalikan kuorum. Untuk informasi lebih lanjut tentang menggunakan AWS Health Dashboard, lihat Panduan Pengguna AWS Health.

Status klaster merah

Status cluster merah berarti bahwa setidaknya satu pecahan primer dan replika tidak dialokasikan ke node. OpenSearch Layanan terus mencoba mengambil snapshot otomatis dari semua indeks terlepas dari statusnya, tetapi snapshot gagal saat status cluster merah tetap ada.

Penyebab paling umum dari status cluster merah adalah node cluster yang gagal dan OpenSearch proses mogok karena beban pemrosesan berat yang terus menerus.

> **Note**
>
> OpenSearch Layanan menyimpan snapshot otomatis selama 14 hari terlepas dari status klaster. Oleh karena itu, jika status klaster merah tetap ada selama lebih dari dua minggu, snapshot otomatis terakhir yang sehat akan dihapus dan Anda bisa secara permanen kehilangan data klaster Anda. Jika domain OpenSearch Layanan Anda memasukkan status klaster merah, AWS Support dapat menghubungi Anda untuk menanyakan apakah Anda ingin mengatasi masalah itu sendiri atau Anda ingin tim dukungan membantu. Anda dapat mengatur CloudWatch alarm untuk memberi tahu Anda ketika status klaster merah terjadi.

Pada akhirnya, pecahan merah menyebabkan gugus merah, dan indeks merah menyebabkan pecahan merah. Untuk mengidentifikasi indeks yang menyebabkan status cluster merah, OpenSearch memiliki beberapa API yang bermanfaat.

- `GET /_cluster/allocation/explain` memilih serpihan pertama yang belum ditetapkan yang ditemukannya dan menjelaskan mengapa hal itu tidak dapat dialokasikan ke simpul:

```
{
    "index": "test4",
    "shard": 0,
    "primary": true,
    "current_state": "unassigned",
    "can_allocate": "no",
    "allocate_explanation": "cannot allocate because allocation is not permitted to
 any of the nodes"
}
```

- `GET /_cat/indices?v` menunjukkan status kesehatan, jumlah dokumen, dan penggunaan disk untuk setiap indeks:

```
health status index           uuid                   pri rep docs.count docs.deleted
 store.size pri.store.size
green  open   test1           30h1EiMvS5uAFr2t5CEVoQ   5   0        820            0
       14mb           14mb
green  open   test2           sdIxs_WDT56afFGu5KPbFQ   1   0          0            0
       233b           233b
green  open   test3           GGRZp_TBRZuSaZpAGk2pmw   1   1          2            0
     14.7kb          7.3kb
red    open   test4           BJxfAErbTtu5HBjIXJV_7A   1   0
green  open   test5           _8C6MIXOSxCqVYicH3jsEA   1   0          7            0
     24.3kb         24.3kb
```

Menghapus indeks merah adalah cara tercepat untuk memperbaiki status cluster merah. Bergantung pada alasan status klaster merah, Anda kemudian dapat menskalakan domain OpenSearch Layanan Anda untuk menggunakan jenis instans yang lebih besar, lebih banyak instance, atau lebih banyak penyimpanan berbasis EBS dan mencoba membuat ulang indeks yang bermasalah.

Jika menghapus indeks bermasalah tidak layak, Anda dapat memulihkan snapshot, menghapus dokumen dari indeks, mengubah pengaturan indeks, mengurangi jumlah replika, atau menghapus indeks lain untuk mengosongkan ruang disk. Langkah penting adalah menyelesaikan status klaster merah sebelum mengonfigurasi ulang domain OpenSearch Layanan Anda. Mengkonfigurasi ulang domain dengan status klaster merah dapat menambah masalah dan menyebabkan domain terjebak dalam status konfigurasi Memproses hingga Anda menyelesaikan status tersebut.

Remediasi otomatis cluster merah

Jika status klaster Anda terus menerus berwarna merah selama lebih dari satu jam, OpenSearch Service mencoba memperbaikinya secara otomatis dengan mengubah rute pecahan yang tidak terisi atau memulihkan dari snapshot sebelumnya.

Jika gagal memperbaiki satu atau lebih indeks merah dan status klaster tetap merah selama total 14 hari, OpenSearch Layanan mengambil tindakan lebih lanjut hanya jika klaster memenuhi setidaknya satu dari kriteria berikut:

- Hanya memiliki satu zona ketersediaan
- Tidak memiliki node master khusus
- Berisi jenis instance burstable (T2 atau T3)

Pada saat ini, jika klaster Anda memenuhi salah satu kriteria ini, OpenSearch Layanan mengirimkan pemberitahuan harian selama 7 hari ke depan yang menjelaskan bahwa jika Anda tidak memperbaiki indeks ini, semua pecahan yang tidak ditetapkan akan dihapus. Jika status klaster Anda masih merah setelah 21 hari, OpenSearch Service menghapus pecahan yang tidak ditetapkan (penyimpanan dan komputasi) pada semua indeks merah. Anda menerima pemberitahuan di panel Pemberitahuan konsol OpenSearch Layanan untuk setiap peristiwa ini. Untuk informasi selengkapnya, lihat the section called “Acara kesehatan cluster”.

Memulihkan dari beban pemrosesan berat yang terus menerus

Untuk menentukan apakah status klaster merah adalah karena beban pemrosesan berat yang terus menerus pada simpul data, pantau metrik klaster berikut.

Metrik terkait	Deskripsi	Pemulihan
JVM MemoryPressure	Tentukan persentase tumpukan Java yang digunakan untuk semua simpul data dalam sebuah klaster. Lihat statistik Maksimum untuk metrik ini, dan cari penurunan tekanan memori yang semakin kecil karena kolektor sampah Java gagal untuk mendapatkan kembali memori yang memadai. Pola ini mungkin disebabkan oleh pengkueri an kompleks atau bidang data yang besar. Tipe instans x86 menggunakan pengumpul sampah Concurrent Mark Sweep (CMS), yang berjalan bersama thread aplikasi untuk membuat jeda tetap singkat. Jika CMS tidak dapat merebut kembali memori yang cukup selama pengumpulan normal, itu memicu pengumpulan sampah penuh, yang dapat menyebabkan jeda aplikasi	Setel pemutus sirkuit memori untuk JVM. Untuk informasi selengkapnya, lihat the section called “JVM OutOfMemoryError”. Jika masalah berlanjut, hapus indeks yang tidak perlu, kurangi jumlah atau kompleksitas permintaa n ke domain, tambahkan instance, atau gunakan jenis instance yang lebih besar.

Metrik terkait	Deskripsi	Pemulihan
	yang lama dan berdampak pada stabilitas cluster. Jenis instance Graviton berbasis ARM menggunakan pengumpul sampah Garbage-First (G1), yang mirip dengan CMS, tetapi menggunakan jeda singkat tambahan dan defragmentasi tumpukan untuk lebih mengurang i kebutuhan pengumpulan sampah penuh. Dalam kedua kasus tersebut, jika penggunaan memori terus tumbuh melampaui apa yang dapat diambil kembali oleh pengumpul sampah selama pengumpulan sampah penuh, OpenSearch macet dengan kesalahan kehabisan memori. Pada semua jenis instance, aturan praktis yang baik adalah menjaga penggunaan di bawah 80%. API `_nodes/stats/jvm` menawarkan ringkasan statistik JVM yang berguna, penggunaan kolam memori, dan informasi pengumpulan sampah: `GET domain-endpoint /_nodes/stats/jvm?pretty`	

Metrik terkait	Deskripsi	Pemulihan
Pemanfaatan CPU	Tentukan persentase sumber daya CPU yang digunakan untuk simpul data dalam sebuah klaster. Lihat statistik Maksimum untuk metrik ini, dan cari pola penggunaan tinggi yang berkelanjutan.	Tambahkan simpul data atau tingkatkan ukuran tipe instans dari simpul data yang ada.
Node	Tentukan jumlah simpul dalam sebuah klaster. Lihat statistik Minimum untuk metrik ini. Nilai ini berfluktuasi ketika layanan men-deploy armada baru dari instans untuk klaster.	Tambahkan simpul data.

Status klaster kuning

Status cluster kuning berarti pecahan utama untuk semua indeks dialokasikan ke node dalam cluster, tetapi pecahan replika untuk setidaknya satu indeks tidak. Cluster simpul tunggal selalu diinisialisasi dengan status klaster kuning karena tidak ada simpul lain yang dapat ditetapkan oleh OpenSearch Service sebagai replika. Untuk mencapai status klaster hijau, tingkatkan jumlah simpul Anda. Untuk informasi selengkapnya, lihat the section called “Mengukur domain”.

Klaster multi-simpul mungkin secara singkat memiliki status klaster kuning setelah membuat indeks baru atau setelah kegagalan simpul. Status ini diselesaikan sendiri sebagai OpenSearch mereplikasi data di seluruh cluster. Kurangnya ruang disk juga dapat menyebabkan status klaster kuning; klaster hanya dapat mendistribusikan serpihan replika jika simpul memiliki ruang disk untuk mengakomodasinya.

ClusterBlockException

Anda mungkin menerima kesalahan `ClusterBlockException` karena sebab-sebab berikut.

Kurangnya ruang penyimpanan yang tersedia

Jika satu atau lebih node di cluster Anda memiliki ruang penyimpanan kurang dari nilai minimum 1) 20% dari ruang penyimpanan yang tersedia, atau 2) 20 GB ruang penyimpanan, operasi penulisan dasar seperti menambahkan dokumen dan membuat indeks dapat mulai gagal. the section called "Menghitung persyaratan penyimpanan"memberikan ringkasan tentang bagaimana OpenSearch Layanan menggunakan ruang disk.

Untuk menghindari masalah, pantau `FreeStorageSpace` metrik di konsol OpenSearch Layanan dan buat CloudWatch alarm untuk dipicu saat `FreeStorageSpace` turun di bawah ambang batas tertentu. `GET /_cat/allocation?v`juga menyediakan ringkasan alokasi shard dan penggunaan disk yang berguna. Untuk mengatasi masalah yang terkait dengan kurangnya ruang penyimpanan, skala domain OpenSearch Layanan Anda untuk menggunakan jenis instans yang lebih besar, lebih banyak instance, atau lebih banyak penyimpanan berbasis EBS.

Tekanan memori JVM tinggi

Ketika MemoryPressure metrik JVM melebihi 92% selama 30 menit, OpenSearch Layanan memicu mekanisme perlindungan dan memblokir semua operasi penulisan untuk mencegah klaster mencapai status merah. Saat perlindungan aktif, operasi penulisan gagal dengan `ClusterBlockException` kesalahan, indeks baru tidak dapat dibuat, dan `IndexCreateBlockException` kesalahan dilemparkan.

Ketika MemoryPressure metrik JVM kembali ke 88% atau lebih rendah selama lima menit, perlindungan dinonaktifkan, dan operasi tulis ke cluster tidak diblokir.

Tekanan memori JVM yang tinggi dapat disebabkan oleh lonjakan jumlah permintaan ke cluster, alokasi pecahan yang tidak seimbang di seluruh node, terlalu banyak pecahan dalam cluster, data lapangan atau ledakan pemetaan indeks, atau jenis instance yang tidak dapat menangani beban masuk. Ini juga dapat disebabkan oleh penggunaan agregasi, wildcard, atau rentang waktu yang luas dalam kueri.

Untuk mengurangi lalu lintas ke cluster dan mengatasi masalah tekanan memori JVM yang tinggi, coba satu atau beberapa hal berikut:

- Skala domain sehingga ukuran heap maksimum per node adalah 32 GB.
- Kurangi jumlah pecahan dengan menghapus indeks lama atau tidak terpakai.
- Hapus cache data dengan operasi `POST` *`index-name`*`/_cache/clear?fielddata=true` API. Perhatikan bahwa membersihkan cache dapat mengganggu kueri yang sedang berlangsung.

Secara umum, untuk menghindari tekanan memori JVM yang tinggi di masa depan, ikuti praktik terbaik berikut:

- Hindari agregasi pada bidang teks, atau ubah jenis pemetaan untuk indeks Anda. `keyword`
- Optimalkan permintaan pencarian dan pengindeksan dengan memilih jumlah pecahan yang benar.
- Siapkan kebijakan Manajemen Negara Indeks (ISM) untuk menghapus indeks yang tidak digunakan secara teratur.

Kesalahan saat bermigrasi ke Multi-AZ dengan Siaga

Masalah berikut mungkin terjadi saat Anda memigrasikan domain yang ada ke Multi-AZ dengan standby.

Membuat indeks, templat indeks, atau kebijakan ISM selama migrasi dari domain tanpa siaga ke domain dengan siaga

Jika Anda membuat indeks saat memigrasikan domain dari Multi-AZ tanpa Standby ke dengan Standby, dan templat indeks atau kebijakan ISM tidak mengikuti pedoman penyalinan data yang disarankan, hal ini dapat menyebabkan inkonsistensi data dan migrasi mungkin gagal. Untuk menghindari situasi ini, buat indeks baru dengan jumlah salinan data (termasuk node primer dan replika) yang merupakan kelipatan dari tiga. Anda dapat memeriksa progres migrasi menggunakan API. `DescribeDomainChangeProgress` Jika Anda menemukan kesalahan jumlah replika, perbaiki kesalahan, lalu hubungi AWSSupport untuk mencoba lagi migrasi.

Jumlah salinan data yang salah

Jika Anda tidak memiliki jumlah salinan data yang tepat di domain Anda, migrasi ke Multi-AZ dengan Siaga akan gagal.

JVM OutOfMemoryError

JVM `OutOfMemoryError` biasanya berarti bahwa salah satu pemutus sirkuit JVM berikut tercapai.

Pemutus sirkuit	Deskripsi	Properti pengaturan klaster
Pemutus Induk	Total persentase memori tumpukan JVM yang	`indices.breaker.total.limit`

Pemutus sirkuit	Deskripsi	Properti pengaturan klaster
	diizinkan untuk semua pemutus sirkuit. Nilai default adalah 95%.	
Pemutus Data Bidang	Persentase memori tumpukan JVM yang diizinkan untuk memuat satu bidang data ke dalam memori. Nilai default adalah 40%. Jika Anda mengunggah data dengan bidang besar, Anda mungkin perlu menaikkan batas ini.	`indices.breaker.fielddata.limit`
Pemutus Permintaan	Persentase memori tumpukan JVM yang diizinkan untuk struktur data yang digunakan untuk menanggapi permintaan layanan. Nilai defaultnya adalah 60%. Jika permintaan layanan Anda melibatkan penghitungan agregasi, Anda mungkin perlu menaikkan batas ini.	`indices.breaker.request.limit`

Simpul klaster yang gagal

Instans Amazon EC2 mungkin mengalami penghentian tak terduga dan memulai ulang. Biasanya, OpenSearch Service me-restart node untuk Anda. Namun, mungkin saja satu atau lebih node dalam sebuah OpenSearch cluster tetap dalam kondisi gagal.

Untuk memeriksa kondisi ini, buka dasbor domain Anda di konsol OpenSearch Layanan. Buka tab Kesehatan cluster dan temukan metrik Total node. Lihat apakah jumlah simpul yang dilaporkan lebih

sedikit dari jumlah yang Anda konfigurasikan untuk klaster Anda. Jika metrik menunjukkan bahwa satu atau lebih simpul mati selama lebih dari satu hari, hubungi AWS Support.

Anda juga dapat mengatur CloudWatch alarm untuk memberi tahu Anda saat masalah ini terjadi.

> **Note**
>
> Metrik Total node tidak akurat selama perubahan konfigurasi klaster Anda dan selama pemeliharaan rutin untuk layanan. Perilaku ini yang diharapkan. Metrik akan melaporkan jumlah simpul klaster yang benar dengan segera. Untuk mempelajari lebih lanjut, lihat the section called “Perubahan konfigurasi”.

Untuk melindungi klaster Anda dari penghentian dan restart node yang tidak terduga, buat setidaknya satu replika untuk setiap indeks di domain Layanan Anda. OpenSearch

Melampaui batas serpihan maksimum

OpenSearch serta 7. x versi Elasticsearch memiliki pengaturan default tidak lebih dari 1.000 pecahan per node. OpenSearch/Elasticsearch memunculkan kesalahan jika permintaan, seperti membuat indeks baru, akan menyebabkan Anda melebihi batas ini. Jika Anda mengalami kesalahan ini, Anda memiliki beberapa pilihan:

- Tambahkan lebih banyak simpul data ke klaster.
- Tingkatkan pengaturan `_cluster/settings/cluster.max_shards_per_node`.
- Gunakan API _shrink untuk mengurangi jumlah serpihan pada simpul.

Domain terjebak dalam status pemrosesan

Domain OpenSearch Layanan Anda memasuki status “Pemrosesan” saat berada di tengah perubahan konfigurasi. Saat Anda memulai perubahan konfigurasi, status domain berubah menjadi “Pemrosesan” sementara OpenSearch Layanan menciptakan lingkungan baru. Di lingkungan baru, OpenSearch Service meluncurkan satu set node baru yang berlaku (seperti data, master, atau UltraWarm). Setelah migrasi selesai, node yang lebih tua dihentikan.

Cluster dapat terjebak dalam status “Pemrosesan” jika salah satu dari situasi ini terjadi:

- Satu set node data baru gagal diluncurkan.

- Migrasi pecahan ke kumpulan node data baru tidak berhasil.
- Pemeriksaan validasi gagal dengan kesalahan.

Untuk langkah-langkah resolusi terperinci dalam setiap situasi ini, lihat Mengapa domain OpenSearch Layanan Amazon saya terjebak dalam status “Pemrosesan”? .

Keseimbangan burst EBS rendah

OpenSearch Layanan mengirimi Anda pemberitahuan konsol ketika saldo burst EBS pada salah satu volume Tujuan Umum (SSD) Anda di bawah 70%, dan pemberitahuan tindak lanjut jika saldo turun di bawah 20%. Untuk memperbaiki masalah ini, Anda dapat meningkatkan skala cluster Anda, atau mengurangi IOPS baca dan tulis sehingga saldo burst dapat dikreditkan. Saldo burst tetap pada 0 untuk domain dengan tipe volume gp3, dan domain dengan volume gp2 yang memiliki ukuran volume di atas 1000 GiB. Untuk informasi selengkapnya, lihat Volume SSD Tujuan Umum (gp2). Anda dapat memantau keseimbangan burst EBS dengan `BurstBalance` CloudWatch metrik.

Tidak dapat mengaktifkan log audit

Anda mungkin mengalami kesalahan berikut saat mencoba mengaktifkan penerbitan log audit menggunakan konsol OpenSearch Layanan:

Kebijakan Akses Sumber Daya yang ditentukan untuk grup CloudWatch log Log tidak memberikan izin yang cukup bagi Amazon OpenSearch Service untuk membuat aliran log. Silakan periksa Kebijakan Akses Sumber Daya.

Jika Anda mengalami kesalahan ini, verifikasi bahwa elemen `resource` dari kebijakan Anda menyertakan grup log ARN yang benar. Jika iya, lakukan langkah-langkah berikut:

1. Tunggu beberapa menit.
2. Segarkan halaman di peramban web Anda.
3. Pilih Pilih grup yang ada.
4. Untuk grup log yang ada, pilih grup log yang Anda buat sebelum menerima pesan galat.
5. Di bagian kebijakan akses, pilih Pilih kebijakan yang ada.
6. Untuk kebijakan yang ada, pilih kebijakan yang Anda buat sebelum menerima pesan galat.
7. Pilih Aktifkan.

Jika kesalahan berlanjut setelah mengulangi proses beberapa kali, hubungi SupportAWS.

Tidak dapat menutup indeks

OpenSearch Layanan mendukung `_close`API hanya untuk OpenSearch dan Elasticsearch versi 7.4 dan yang lebih baru. Jika Anda menggunakan versi yang lebih lama dan memulihkan indeks dari snapshot, Anda dapat menghapus indeks yang ada (sebelum atau setelah mengindeks ulang itu).

Periksa lisensi klien

Distribusi default Logstash dan Beats mencakup pemeriksaan lisensi berpemilik dan gagal terhubung ke versi open source. OpenSearch Pastikan Anda menggunakan distribusi Apache 2.0 (OSS) dari klien ini dengan Layanan. OpenSearch

Permintaan throttling

Jika Anda menerima kesalahan `403 Request throttled due to too many requests` atau `429 Too Many Requests` terus-menerus, pertimbangkan untuk menskalakan secara vertikal. Amazon OpenSearch Service membatasi permintaan jika payload akan menyebabkan penggunaan memori melebihi ukuran maksimum heap Java.

Tidak dapat SSH ke simpul

Anda tidak dapat menggunakan SSH untuk mengakses salah satu node di OpenSearch cluster Anda, dan Anda tidak dapat langsung memodifikasi`opensearch.yml`. Sebaliknya, gunakan konsol, AWS CLI, atau SDK untuk mengonfigurasi domain Anda. Anda juga dapat menentukan beberapa pengaturan tingkat cluster menggunakan OpenSearch REST API. Untuk mempelajari selengkapnya, lihat Referensi API OpenSearch Layanan Amazon danthe section called “Operasi yang didukung”.

Jika Anda membutuhkan lebih banyak wawasan tentang kinerja cluster, Anda dapat mempublikasikan log kesalahan dan memperlambat log CloudWatch.

Kesalahan snapshot “Tidak Valid untuk Kelas Penyimpanan Objek”

OpenSearch Snapshot layanan tidak mendukung kelas penyimpanan S3 Glacier. Anda mungkin mengalami kesalahan ini ketika Anda mencoba untuk membuat daftar snapshot jika bucket S3 Anda menyertakan aturan siklus hidup yang mentransisikan objek ke kelas penyimpanan S3 Glacier.

Jika Anda perlu memulihkan snapshot dari bucket, pulihkan objek dari S3 Glacier, salin objek ke bucket baru, dan daftarkan bucket baru sebagai repositori snapshot.

Header host tidak valid

OpenSearch Layanan mengharuskan klien menentukan `Host` di header permintaan. Nilai `Host` yang valid adalah titik akhir domain tanpa `https://`, seperti:

```
Host: search-my-sample-domain-ih2lhn2ew2scurji.us-west-2.es.amazonaws.com
```

Jika Anda menerima `Invalid Host Header` kesalahan saat membuat permintaan, periksa apakah klien atau proxy Anda menyertakan titik akhir domain OpenSearch Layanan (dan bukan, misalnya, alamat IP-nya) di `Host` header.

Tipe instans M3 tidak valid

OpenSearch Layanan tidak mendukung penambahan atau modifikasi instance M3 ke domain yang ada yang berjalan OpenSearch atau Elasticsearch versi 6.7 dan yang lebih baru. Anda dapat terus menggunakan instans M3 dengan Elasticsearch 6.5 dan yang lebih lama.

Sebaiknya pilih tipe instans yang lebih baru. Untuk domain yang berjalan OpenSearch atau Elasticsearch 6.7 atau yang lebih baru, pembatasan berikut berlaku:

- Jika domain Anda yang ada tidak menggunakan instans M3, Anda tidak dapat lagi mengubahnya.
- Jika Anda mengubah domain yang ada dari tipe instans M3 ke tipe instans lain, Anda tidak dapat beralih kembali.

Kueri panas berhenti berfungsi setelah mengaktifkan UltraWarm

Saat Anda mengaktifkan UltraWarm pada domain, jika tidak ada penggantian yang sudah ada sebelumnya ke `search.max_buckets` pengaturan, OpenSearch Service secara otomatis menetapkan nilainya `10000` untuk mencegah kueri yang berat memori menjenuhkan node hangat. Jika kueri panas Anda menggunakan lebih dari 10.000 bucket, kueri tersebut mungkin berhenti berfungsi saat Anda mengaktifkan. UltraWarm

Karena Anda tidak dapat mengubah setelan ini karena sifat Amazon OpenSearch Service yang dikelola, Anda perlu membuka kasus dukungan untuk meningkatkan batas. Peningkatan batas tidak memerlukan langganan dukungan premium.

Tidak dapat menurunkan versi setelah peningkatan

Upgrade di tempat tidak dapat dibatalkan, tetapi jika Anda menghubungi AWS Support, mereka dapat membantu Anda memulihkan snapshot pra-peningkatan otomatis pada domain baru. Misalnya, jika Anda meningkatkan domain dari Elasticsearch 5.6 ke 6.4, AWS Support dapat membantu Anda memulihkan snapshot pra-peningkatan pada domain Elasticsearch 5.6 baru. Jika Anda mengambil snapshot manual dari domain asli, Anda dapat melakukan langkah itu sendiri.

Perlu ringkasan domain untuk semua Wilayah AWS

Skrip berikut menggunakan AWS CLIperintah mendeskripsikan wilayah Amazon EC2 untuk membuat daftar semua Wilayah di mana OpenSearch Layanan dapat tersedia. Kemudian menyerukan list-domain-namesuntuk setiap Wilayah:

```
for region in `aws ec2 describe-regions --output text | cut -f4`
do
    echo "\nListing domains in region '$region':"
    aws opensearch list-domain-names --region $region --query 'DomainNames'
done
```

Anda menerima output berikut untuk setiap Wilayah:

```
Listing domains in region:'us-west-2'...
[
  {
    "DomainName": "sample-domain"
  }
]
```

Wilayah di mana OpenSearch Layanan tidak tersedia mengembalikan “Tidak dapat terhubung ke URL titik akhir.”

Kesalahan browser saat menggunakan OpenSearch Dasbor

Browser Anda membungkus pesan kesalahan layanan dalam objek respons HTTP saat Anda menggunakan Dasbor untuk melihat data di domain OpenSearch Layanan Anda. Anda dapat menggunakan alat developer yang umumnya tersedia di peramban web, seperti Mode Developer

di Chrome, untuk melihat kesalahan layanan yang mendasarinya dan membantu upaya debugging Anda.

Untuk melihat kesalahan layanan di Chrome

1. Dari bilah menu atas Chrome, pilih Lihat, Pengembang, Alat Pengembang.
2. Pilih tab Jaringan.
3. Di kolom Status, pilih sesi HTTP apa pun dengan status 500.

Untuk melihat kesalahan layanan di Firefox

1. Dari menu, pilih Alat, Developer Web, Jaringan.
2. Pilih sesi HTTP apa pun dengan status 500.
3. Pilih tab Respons untuk melihat respons layanan.

Pecahan simpul dan kemiringan penyimpanan

Node shard skew adalah ketika satu atau lebih node dalam sebuah cluster memiliki pecahan yang jauh lebih banyak daripada node lainnya. Skew penyimpanan node adalah ketika satu atau lebih node dalam sebuah cluster memiliki storage (`disk.indices`) secara signifikan lebih banyak daripada node lainnya. Meskipun kedua kondisi ini dapat terjadi sementara, seperti ketika domain telah menggantikan node dan masih mengalokasikan pecahan untuk itu, Anda harus mengatasinya jika tetap ada.

Untuk mengidentifikasi kedua jenis kemiringan, jalankan operasi _cat/allocation API dan bandingkan entri dan entri dalam `shards` respons: `disk.indices`

```
shards    | disk.indices  | disk.used  | disk.avail   | disk.total   | disk.percent |
host     | ip       | node
   264    |      465.3mb  |   229.9mb  |      1.4tb   |      1.5tb   |            0 |
 x.x.x.x  | x.x.x.x  | node1
   115    |        7.9mb  |    83.7mb  |     49.1gb   |     49.2gb   |            0 |
x.x.x.x  | x.x.x.x  | node2
   264    |      465.3mb  |   235.3mb  |      1.4tb   |      1.5tb   |            0 |
 x.x.x.x  | x.x.x.x  | node3
   116    |        7.9mb  |    82.8mb  |     49.1gb   |     49.2gb   |            0 |
x.x.x.x  | x.x.x.x  | node4
```

```
    115     |        8.4mb   |      85mb   |     49.1gb    |     49.2gb    |            0 |
 x.x.x.x   | x.x.x.x   | node5
```

Sementara beberapa kemiringan penyimpanan adalah normal, apa pun yang lebih dari 10% dari rata-rata adalah signifikan. Ketika distribusi shard miring, penggunaan CPU, jaringan, dan bandwidth disk juga bisa menjadi miring. Karena lebih banyak data umumnya berarti lebih banyak operasi pengindeksan dan pencarian, node terberat juga cenderung menjadi node yang paling tegang sumber daya, sedangkan node yang lebih ringan mewakili kapasitas yang kurang dimanfaatkan.

Remediasi: Gunakan jumlah pecahan yang merupakan kelipatan dari jumlah node data untuk memastikan bahwa setiap indeks didistribusikan secara merata di seluruh node data.

Pecahan indeks dan kemiringan penyimpanan

Index shard skew adalah ketika satu atau lebih node memegang lebih banyak pecahan indeks daripada node lainnya. Kemiringan penyimpanan indeks adalah ketika satu atau lebih node menyimpan jumlah penyimpanan total indeks yang tidak proporsional.

Kemiringan indeks lebih sulit diidentifikasi daripada kemiringan simpul karena memerlukan beberapa manipulasi output API _cat/shards. Selidiki kemiringan indeks jika ada beberapa indikasi kemiringan dalam metrik cluster atau node. Berikut ini adalah indikasi umum dari kemiringan indeks:

- Kesalahan HTTP 429 terjadi pada subset node data
- Indeks tidak merata atau operasi pencarian antrian di seluruh node data
- Tumpukan JVM dan/atau pemanfaatan CPU yang tidak merata di seluruh node data

Remediasi: Gunakan jumlah pecahan yang merupakan kelipatan dari jumlah node data untuk memastikan bahwa setiap indeks didistribusikan secara merata di seluruh node data. Jika Anda masih melihat penyimpanan indeks atau shard miring, Anda mungkin perlu memaksa realokasi pecahan, yang terjadi dengan setiap penerapan biru/hijau domain Layanan Anda. OpenSearch

Operasi yang tidak sah setelah memilih akses VPC

Saat Anda membuat domain baru menggunakan konsol OpenSearch Layanan, Anda memiliki opsi untuk memilih VPC atau akses publik. Jika Anda memilih akses VPC, OpenSearch Layanan kueri untuk informasi VPC dan gagal jika Anda tidak memiliki izin yang tepat:

```
You are not authorized to perform this operation. (Service: AmazonEC2; Status Code:
 403; Error Code: UnauthorizedOperation
```

Untuk mengaktifkan kueri ini, Anda harus memiliki akses ke operasi `ec2:DescribeVpcs`, `ec2:DescribeSubnets`, dan `ec2:DescribeSecurityGroups`. Persyaratan ini hanya untuk konsol. Jika Anda menggunakan AWS CLI untuk membuat dan mengonfigurasi domain dengan VPC endpoint, Anda tidak memerlukan akses ke operasi tersebut.

Terjebak saat memuat setelah membuat domain VPC

Setelah membuat domain baru yang menggunakan akses VPC, Status konfigurasi domain mungkin tidak akan pernah mengalami kemajuan melampaui Pemuatan. Jika masalah ini terjadi, Anda mungkin telah menonaktifkan AWS Security Token Service (AWS STS) untuk Wilayah Anda.

Untuk menambahkan titik akhir VPC ke VPC Anda, OpenSearch Layanan perlu mengambil peran tersebut. `AWSServiceRoleForAmazonOpenSearchService` Dengan demikian, AWS STS harus diaktifkan untuk membuat domain baru yang menggunakan akses VPC di Wilayah tertentu. Untuk mempelajari selengkapnya tentang mengaktifkan dan menonaktifkan AWS STS, lihat Panduan Pengguna IAM.

Menolak permintaan ke OpenSearch API

Dengan diperkenalkannya kontrol akses berbasis tag untuk OpenSearch API, Anda mungkin mulai melihat kesalahan akses ditolak yang sebelumnya tidak Anda lihat. Ini mungkin karena satu atau beberapa kebijakan akses Anda berisi `Deny` penggunaan `ResourceTag` kondisi, dan kondisi tersebut sekarang sedang dihormati.

Misalnya, kebijakan berikut digunakan untuk hanya menolak akses ke `CreateDomain` tindakan dari API konfigurasi, jika domain memiliki tag`environment=production`. Meskipun daftar tindakan juga termasuk`ESHttpPut`, pernyataan penolakan tidak berlaku untuk tindakan itu atau `ESHttp*` tindakan lainnya.

```
{
  "Version": "2012-10-17",
  "Statement": [{
    "Action": [
      "es:CreateDomain",
      "es:ESHttpPut"
```

```
    ],
    "Effect": "Deny",
    "Resource": "*",
    "Condition": {
      "ForAnyValue:StringEquals": {
        "aws:ResourceTag/environment": [
          "production"
        ]
      }
    }
  }]
}
```

Dengan dukungan tambahan tag untuk metode OpenSearch HTTP, kebijakan berbasis identitas IAM seperti di atas akan mengakibatkan pengguna terlampir ditolak akses ke tindakan. `ESHttpPut` Sebelumnya, dengan tidak adanya validasi tag, pengguna terlampir masih dapat mengirim permintaan PUT.

Jika Anda mulai melihat kesalahan akses ditolak setelah memperbarui domain Anda ke perangkat lunak layanan R20220323 atau yang lebih baru, periksa kebijakan akses berbasis identitas Anda untuk melihat apakah ini masalahnya dan perbarui jika perlu untuk mengizinkan akses.

Tidak dapat terhubung dari Alpine Linux

Alpine Linux membatasi ukuran respons DNS ke 512 byte. Jika Anda mencoba untuk terhubung ke domain OpenSearch Layanan Anda dari Alpine Linux versi 3.18.0 atau lebih rendah, resolusi DNS dapat gagal jika domain berada dalam VPC dan memiliki lebih dari 20 node. Jika Anda menggunakan versi Alpine Linux yang lebih tinggi dari 3.18.0, Anda harus dapat menyelesaikan lebih dari 20 host. Untuk informasi selengkapnya, lihat catatan rilis Alpine Linux 3.18.0.

Jika domain Anda berada dalam VPC, sebaiknya gunakan distribusi Linux lainnya, seperti Debian, Ubuntu, CentOS, Red Hat Enterprise Linux, atau Amazon Linux 2, untuk menghubungkannya.

Terlalu banyak permintaan untuk Search Backpressure

Kontrol penerimaan berbasis CPU adalah mekanisme penjaga gerbang yang secara proaktif membatasi jumlah permintaan ke node berdasarkan kapasitasnya saat ini, baik untuk peningkatan organik maupun lonjakan lalu lintas. Permintaan yang berlebihan mengembalikan kode status HTTP 429 “Terlalu Banyak Permintaan” setelah ditolak. Kesalahan ini menunjukkan sumber daya klaster

yang tidak mencukupi, permintaan pencarian intensif sumber daya, atau lonjakan beban kerja yang tidak diinginkan.

Search Backpressure memberikan alasan penolakan, yang dapat membantu menyempurnakan permintaan pencarian intensif sumber daya. Untuk lonjakan lalu lintas, kami merekomendasikan percobaan ulang sisi klien dengan backoff dan jitter eksponensial.

Kesalahan sertifikat saat menggunakan SDK

Karena AWS SDK menggunakan sertifikat CA dari komputer Anda, perubahan pada sertifikat di server AWS dapat menyebabkan kegagalan koneksi saat Anda mencoba menggunakan SDK. Pesan kesalahan bervariasi, tetapi biasanya berisi teks berikut:

```
Failed to query OpenSearch
...
SSL3_GET_SERVER_CERTIFICATE:certificate verify failed
```

Anda dapat mencegah kegagalan ini dengan menyimpan sertifikat CA dan sistem operasi komputer Anda up-to-date. Jika Anda mengalami masalah ini di lingkungan perusahaan dan tidak mengelola komputer Anda sendiri, Anda mungkin perlu meminta administrator untuk membantu proses pembaruan.

Daftar berikut menunjukkan sistem operasi minimum dan versi Java:

- Versi Microsoft Windows yang memiliki pembaruan mulai Januari 2005 atau yang lebih baru yang sudah terinstal memuat setidaknya satu dari CA yang diperlukan dalam daftar kepercayaan mereka.
- Mac OS X 10.4 dengan Java untuk Mac OS X 10.4 Release 5 (Februari 2007), Mac OS X 10.5 (Oktober 2007), dan versi lebih baru memuat setidaknya satu dari CA yang diperlukan dalam daftar kepercayaan mereka.
- Red Hat Enterprise Linux 5 (Maret 2007), 6, dan 7 serta CentOS 5, 6, dan 7 semuanya berisi setidaknya satu dari CA yang diperlukan dalam daftar default CA tepercaya mereka.
- Java 1.4.2_12 (Mei 2006), 5 Pembaruan 2 (Maret 2005), dan semua versi setelahnya, termasuk Java 6 (Desember 2006), 7, dan 8, memuat setidaknya satu dari CA yang diperlukan dalam daftar default CA tepercaya mereka.

Ketiga otoritas sertifikasi (CA) adalah:

- Amazon Root CA 1
- Starfield Services Root Certificate Authority - G2
- Starfield Class 2 Certification Authority

Sertifikat root dari dua otoritas pertama tersedia dari Amazon Trust Services, tetapi menjaga komputer Anda up-to-date adalah solusi yang lebih mudah. Untuk mempelajari lebih lanjut tentang sertifikat yang disediakan ACM, lihat AWS Certificate Manager FAQ.

> **Note**
>
> Saat ini, domain OpenSearch Layanan di Wilayah us-timur-1 menggunakan sertifikat dari otoritas yang berbeda. Kami berencana untuk memperbarui Wilayah untuk menggunakan otoritas sertifikat baru ini dalam waktu dekat.

Riwayat dokumen untuk OpenSearch Layanan Amazon

Topik ini menjelaskan perubahan penting pada OpenSearch Layanan Amazon. Pembaruan perangkat lunak layanan menambahkan dukungan untuk fitur baru, patch keamanan, perbaikan bug, dan peningkatan lainnya. Untuk menggunakan fitur baru, Anda mungkin perlu memperbarui perangkat lunak layanan di domain Anda. Untuk informasi selengkapnya, lihat the section called "Pembaruan perangkat lunak layanan".

Fitur layanan diluncurkan secara bertahap ke Wilayah AWS tempat layanan tersedia. Kami memperbarui dokumentasi ini hanya untuk rilis pertama. Kami tidak memberikan informasi tentang ketersediaan Wilayah atau mengumumkan peluncuran Wilayah berikutnya. Untuk informasi tentang ketersediaan fitur layanan wilayah, dan untuk berlangganan pemberitahuan tentang pembaruan, lihat Apa yang Baru dengan AWS?

Tanggal yang relevan dengan sejarah ini:

- Versi produk saat ini— 2021-01-01
- Rilis produk terbaru— 1 April 2024
- Pembaruan dokumentasi terbaru- April 1, 2024

Untuk pembaruan mengenai pembaruan, Anda dapat berlangganan ke umpan RSS.

> **Note**
>
> Rilis patch: Versi perangkat lunak layanan yang diakhiri dengan "-P" dan nomor, seperti R20211203-P4, adalah rilis patch. Patch cenderung mencakup peningkatan kinerja, perbaikan bug kecil, dan perbaikan keamanan atau perbaikan postur. Karena tambalan tidak menyertakan fitur baru atau perubahan yang melanggar, mereka umumnya tidak memiliki dampak pengguna atau dokumentasi langsung, itulah sebabnya spesifikasi setiap tambalan tidak termasuk dalam riwayat dokumen ini.

Perubahan	Deskripsi	Tanggal
Dukungan Amazon OpenSearch Ingestion untuk Data Prepper versi 2.7	Amazon OpenSearch Ingestion menambahkan dukungan untuk Data Prepper	April 4, 2024

	versi 2.7. Untuk informasi lebih lanjut, lihat catatan rilis 2.7.	
Layanan AWS akses pribadi untuk koleksi OpenSearch Tanpa Server	Anda sekarang dapat memberikan akses khusus Layanan AWS, seperti Amazon Bedrock, ke koleksi OpenSearch Tanpa Server Anda dalam kebijakan akses jaringan.	Maret 28, 2024
Pembaruan EBS di tempat	Anda sekarang dapat membuat beberapa perubahan EBS ke domain Anda tanpa penerapan biru/hijau di Amazon Service. OpenSearch	Februari 14, 2024
Konfigurasi mengubah visibilitas	Anda sekarang dapat melacak perubahan konfigurasi domain di konsol OpenSearch Layanan Amazon dan menggunakan API konfigurasi.	Februari 6, 2024
Koleksi pencarian vektor ketersediaan umum	Koleksi pencarian vektor Amazon OpenSearch Tanpa Server sekarang tersedia secara umum. Perbaikan penting berikut dilakukan selama fase pratinjau: • Koleksi pencarian vektor sekarang mendukung beban kerja dengan miliaran vektor, masing-masing dengan hingga 128 dimensi. • OpenSearch Dasbor sekarang mendukung koleksi pencarian vektor.	November 29, 2023

Contoh OR1	Amazon OpenSearch Service sekarang mendukung jenis instans OR1.	November 29, 2023
Kueri langsung dengan Amazon S3 (pratinjau)	Kueri langsung memberikan solusi yang dikelola sepenuhnya untuk membuat data transaksional tersedia di OpenSearch Layanan Amazon dalam hitungan detik setelah ditulis ke bucket Amazon S3.	November 29, 2023
Kapasitas 10 TiB untuk koleksi time series	Amazon OpenSearch Serverless menambahkan dukungan hingga 10 TiB data indeks untuk koleksi deret waktu. Rilis ini juga mendukung kapasitas maksimum 200 OCU untuk semua jenis koleksi dan kemampuan untuk menonaktifkan replika siaga saat Anda membuat koleksi.	November 29, 2023
OpenSearch 2.11 dukungan	Amazon OpenSearch Service sekarang mendukung OpenSearch versi 2.11. Versi ini mencakup semua fitur yang merupakan bagian dari versi 2.10 dan 2.11. Untuk informasi lebih lanjut, lihat catatan rilis 2.10 dan 2.11.	17 November 2023

Dukungan Amazon OpenSearch Ingestion untuk Data Prepper versi 2.6	Amazon OpenSearch Ingestion menambahkan dukungan untuk Data Prepper versi 2.6. Untuk informasi selengkapnya, lihat catatan rilis 2.6. Selain itu, Anda dapat menentukan Amazon DynamoDB sebagai sumber pipeline. Untuk informasi selengkapnya, lihat Menggunakan pipeline OpenSearch Ingestion dengan Amazon DynamoDB.	17 November 2023
Dukungan Amazon OpenSearch Ingestion untuk Data Prepper versi 2.5	Amazon OpenSearch Ingestion menambahkan dukungan untuk Data Prepper versi 2.5. Untuk informasi lebih lanjut, lihat catatan rilis 2.5. Selain itu, Anda sekarang dapat menentukan domain OpenSearch Layanan atau koleksi OpenSearch Tanpa Server sebagai sumber pipeline. Untuk informasi selengkapnya, lihat plugin OpenSearch sumber di dokumen Data Prepper.	17 November 2023

CloudFormation template untuk inferensi jarak jauh	Untuk memudahkan penyiapan inferensi jarak jauh untuk penelusuran semantik, Amazon OpenSearch Service menyediakan AWS CloudFormation template di konsol yang mengotomatiskan proses penyediaan model untuk Anda.	7 November 2023
Pembaruan ke kebijakan peran terkait layanan	Menambahkan izin yang diperlukan untuk kebijakan peran terkait layanan untuk menetapkan dan membatalkan penetapan `AmazonOpenSearchServiceRolePolicy` alamat IPv6. Kebijakan Elasticsearch yang tidak digunakan lagi juga `AmazonElasticsearchServiceRolePolicy` telah diperbarui untuk memastikan kompatibilitas mundur.	26 Oktober 2023

Kebijakan siklus OpenSearch hidup Amazon Tanpa Server	Amazon OpenSearch Serverless memperkenalkan kebijakan siklus hidup indeks untuk merampingkan pengelolaan penyimpanan dan penghapusan data. Sekarang Anda dapat menggunakan API atau antarmuka konfigurasi di konsol untuk mengatur kebijakan retensi data untuk koleksi deret waktu, sehingga tidak perlu membuat indeks atau skrip harian untuk menghapus data lama.	25 Oktober 2023
Dukungan instans IM4gn	Amazon OpenSearch Service sekarang mendukung tipe instans iM4gn. Instans IM4GN dioptimalkan untuk beban kerja yang mengelola kumpulan data besar dan membutuhkan kepadatan penyimpanan yang tinggi per vCPU.	20 Oktober 2023

Opsi administratif	Amazon OpenSearch Service sekarang menawarkan beberapa opsi administratif yang menyediakan kontrol terperinci jika Anda perlu memecahkan masalah dengan domain Anda. Pilihan ini termasuk kemampuan untuk memulai kembali OpenSearch proses pada node data dan kemampuan untuk me-restart node data.	17 Oktober 2023
Plugin opsional	Amazon OpenSearch Service menambahkan dukungan untuk empat plugin penganalisis bahasa baru: Nori (Korea), Sudachi (Jepang), Pinyin (Mandarin), dan Analisis STConvert (Mandarin), serta plugin Amazon Personalize Search Ranking.	16 Oktober 2023
OpenSearch 2.9 dukungan	Amazon OpenSearch Service sekarang mendukung OpenSearch versi 2.9. Versi ini mencakup semua fitur yang merupakan bagian dari versi 2.8 dan 2.9. Untuk informasi lebih lanjut, lihat catatan rilis 2.8 dan 2.9.	2 Oktober 2023

Konektor ML	Amazon OpenSearch Service menambahkan dukungan untuk konektor machine learning (ML). Konektor memfasilitasi akses ke model ML yang di-host di platform lain Layanan AWS, atau pada platform machine learning (ML) pihak ketiga.	September 6, 2023
Amazon OpenSearch Ingestion menambahkan dukungan untuk Data Prepper versi 2.4	Amazon OpenSearch Ingestion menambahkan dukungan untuk Data Prepper versi 2.4. Untuk informasi selengkapnya, lihat catatan rilis 2.4. Selain itu, Anda sekarang dapat menentukan Amazon Managed Streaming for Apache Kafka (Amazon MSK) sebagai sumber pipa.	31 Agustus 2023
Kapasitas 6 TiB untuk koleksi time series	Amazon OpenSearch Serverless menambahk an dukungan hingga 6 TiB data indeks untuk koleksi deret waktu. Rilis ini juga mendukung kapasitas maksimum yang diizinkan 100 OCU untuk koleksi pencarian dan seri waktu.	15 Agustus 2023

Koleksi pencarian vektor	Amazon OpenSearch Serverless menambahkan opsi untuk membuat koleksi pencarian vektor, yang dapat Anda gunakan untuk menyimpan penyematan vektor untuk mendukung pencarian persamaan dan semantik.	26 Juli 2023
OpenSearch 2.7 dukungan	Amazon OpenSearch Service sekarang mendukung OpenSearch versi 2.7. Versi ini mencakup semua fitur yang merupakan bagian dari versi 2.6 dan 2.7. Untuk informasi lebih lanjut, lihat catatan rilis 2.6 dan 2.7.	Juli 10, 2023
Dukungan Data Prepper 2.3	Amazon OpenSearch Ingestion menambahkan dukungan Data Prepper versi 2.3. Untuk informasi selengkapnya, lihat catatan rilis 2.3. Selain itu, Anda sekarang dapat menentukan Amazon Security Lake sebagai sumber pipa.	26 Juni 2023

Multi-AZ dengan Siaga	Amazon OpenSearch Service menambahkan opsi untuk menerapkan domain di tiga Availability Zone (AZ), dengan masing-masing AZ berisi salinan data lengkap dan dengan node di salah satu AZ ini bertindak sebagai siaga. Opsi penyebaran Multi-AZ dengan Siaga menyediakan ketersediaan 99,99% dan kinerja yang konsisten jika terjadi kegagalan infrastruktur.	3 Mei 2023
Peran terkait layanan baru	Amazon OpenSearch Service menambahkan peran terkait layanan yang disebut`AWSServiceRoleForAmazonOpenSearchIngestion` , yang memungkinkan Amazon OpenSearch Ingestion mengirim data metrik ke. Amazon CloudWatch	April 26, 2023

OpenSearch Tertelan Amazon	Amazon OpenSearch Ingestion adalah pengumpul data terkelola penuh yang mengirimkan data log dan jejak waktu nyata ke domain OpenSearch Layanan dan koleksi Tanpa Server. OpenSearch OpenSearch Ingestion menghilangkan kebutuhan bagi Anda untuk menggunakan solusi pihak ketiga seperti Logstash atau Jaeger untuk menelan data ke dalam domain dan koleksi Anda.	April 26, 2023
OpenSearch 2.5 dukungan	Amazon OpenSearch Service sekarang mendukung OpenSearch versi 2.5. Versi ini mencakup semua fitur yang merupakan bagian dari versi 2.4 dan 2.5. Untuk informasi selengkapnya, lihat catatan rilis 2.4 dan 2.5.	13 Maret 2023

Jendela perawatan di luar puncak	Amazon OpenSearch Service menambahkan jendela off-peak, yang merupakan blok waktu lalu lintas rendah 10 jam setiap hari di mana ia dapat menjadwalkan pembaruan perangkat lunak layanan dan optimasi Auto-Tune yang memerlukan penerapan biru/hijau. Pembaruan off-peak membantu meminimalkan ketegangan pada node master khusus cluster selama periode lalu lintas yang lebih tinggi. Untuk domain baru yang dibuat setelah 16 Februari, jendela off-peak secara otomatis dikonfigurasi antara pukul 10:00 hingga 8:00 pagi waktu setempat. Untuk domain yang ada, Anda perlu mengaktifkan jendela secara eksplisit.	16 Februari 2023
Konfigurasikan otentikasi SAMB selama pembuatan domain	Amazon OpenSearch Service sekarang mendukung konfigurasi otentikasi SAMB selama pembuatan domain. Sebelumnya, Anda harus mengkonfigurasi opsi SAFL setelah domain sudah dibuat.	1 Februari 2023

Remote reindex untuk domain VPC	Amazon OpenSearch Service menambahkan opsi untuk koneksi titik akhir VPC antara dua domain. Anda sekarang dapat menggunakan reindex jarak jauh untuk menyalin indeks dari satu domain VPC ke domain VPC lainnya tanpa proxy terbalik. Domain VPC Anda harus menjalankan perangkat lunak layanan R20221114 atau yang lebih baru untuk menggunakan fitur ini.	31 Januari 2023

Ketersediaan OpenSearch umum Amazon Tanpa Server	Amazon OpenSearch Serverless sekarang tersedia secara umum. Perbaikan penting berikut dilakukan selama fase pratinjau: • Kapasitas sekarang dapat diturunkan ke OCU minimum yang dikonfigu rasi ketika ada penurunan lalu lintas pada titik akhir pengumpulan. • OCU maksimum yang diizinkan untuk pengindek san dan pencarian ditingkat kan dari 20 menjadi 50. Setiap OCU mencakup penyimpanan singkat panas yang cukup untuk 120 GiB data indeks. • Anda sekarang dapat mengonfigurasi pengatura n akses data saat membuat koleksi, daripada harus mengonfigurasinya dalam alur kerja terpisah.	Januari 25, 2023

Lari kering asinkron	Amazon OpenSearch Service sekarang mendukung async dry run, yang memungkinkan Anda melakukan pemeriksa an validasi sebelum membuat perubahan konfigurasi, dan memberi tahu Anda jika perubahan Anda akan menyebabkan penerapan biru/ hijau.	19 Januari 2023
Peran terkait layanan baru	Amazon OpenSearch Service menambahkan peran terkait layanan yang disebut`AWSServic eRoleForAmazonOpen SearchServerless` , yang memungkinkan OpenSearch Tanpa Server mengirim data metrik ke. Amazon CloudWatc h	29 November 2022
OpenSearch Pratinjau Amazon Tanpa Server	Amazon OpenSearch Serverless adalah konfigura si on-demand, penskalaan otomatis, tanpa server untuk Amazon Service. OpenSearc h Tanpa server menghilan gkan kompleksitas operasion al penyediaan, konfigurasi, dan penyetelan cluster Anda. OpenSearch	29 November 2022

OpenSearch 2.3 dukungan	Amazon OpenSearch Service sekarang mendukung OpenSearch versi 2.3. Versi ini mencakup semua fitur yang merupakan bagian versi 2.0, 2.1, dan 2.2. Untuk informasi selengkapnya, lihat catatan rilis 2.0, 2.1, 2.2, dan 2.3. Versi 2.3 berisi perubahan yang melanggar. Untuk informasi selengkapnya, lihat Jalur pemutakhiran yang didukung.	15 November 2022
Pemberitahuan dukungan plugin	Amazon OpenSearch Service sekarang mendukung plugin Notifications, yang menawarkan lokasi pusat untuk semua notifikasi Anda dari OpenSearch plugin. Dimulai dengan versi 2.0, tujuan peringatan tidak digunakan lagi dan diganti dengan saluran notifikasi.	15 November 2022

Dukungan Kibana 7.1.1	Domain Amazon OpenSearch Service yang menjalankan Elasticsearch 7.1 sekarang mendukung rilis patch terbaru untuk Kibana 7.1.1, yang menambahkan perbaikan bug dan meningkatkan keamanan. Saat Anda memperbarui domain 7.1 Anda ke perangkat lunak layanan R20221114, OpenSearch Layanan akan secara otomatis memutakhirkannya ke rilis patch ini.	15 November 2022
Dukungan Kibana 6.8.13	Domain Amazon OpenSearch Service yang menjalankan Elasticsearch 6.8 sekarang mendukung rilis patch terbaru untuk Kibana 6.8.13, yang menambahkan perbaikan bug dan meningkatkan keamanan. Saat Anda memperbarui domain 6,8 Anda ke perangkat lunak layanan R20221114, OpenSearch Layanan akan secara otomatis memutakhirkannya ke rilis patch ini.	15 November 2022

Dukungan Kibana 6.3.2	Domain Amazon OpenSearch Service yang menjalankan Elasticsearch 6.3 sekarang mendukung rilis patch terbaru untuk Kibana 6.3.2, yang menambahkan perbaikan bug dan meningkatkan keamanan. Saat Anda memperbarui domain 6.3 Anda ke perangkat lunak layanan R20221114, OpenSearch Layanan akan secara otomatis memutakhirkannya ke rilis patch ini.	15 November 2022

AWS PrivateLink	Dengan titik akhir OpenSearch VPC yang dikelola Layanan Amazon, Anda dapat terhubung langsung ke domain OpenSearch VPC Layanan dengan menggunakan titik akhir VPC antarmuka alih-alih menghubungkan melalui internet. Titik akhir OpenSearch VPC yang dikelola Layanan hanya dapat diakses dalam VPC tempat titik akhir disediakan, atau dari VPC mana pun yang diintip dengan VPC tempat titik akhir disediakan, sebagaimana diizinkan oleh tabel rute dan grup keamanan. Domain VPC Anda harus menjalankan perangkat lunak layanan R20220928 atau yang lebih baru untuk terhubung ke titik akhir VPC antarmuka.	7 November 2022
Perbaikan bug dan peningkatan kinerja	Perangkat lunak layanan R20220928 mencakup perbaikan bug dan peningkatan kinerja, termasuk pencatatan SAFL yang ditingkatkan. Pembaruan juga mengubah penyewa default menjadi `Global` bukan`Private`.	3 Oktober 2022

Referensi API yang ditingkatkan	Amazon OpenSearch Service menawarkan referensi API konfigurasi yang ditingkatkan dan mencakup semua. Referensi baru berisi semua tindakan dan tipe data yang tersedia, contoh permintaan dan sintaks respons, dan tautan ke referensi SDK yang sesuai untuk semua bahasa yang didukung.	13 September 2022
Validasi biru/hijau	Amazon OpenSearch Service sekarang melakukan pemeriksaan validasi sebelum penerapan biru/hijau, dan menampilkan kesalahan validasi jika domain Anda tidak memenuhi syarat untuk pembaruan.	Agustus 16, 2022
OpenSearch 1.3 dukungan	Amazon OpenSearch Service sekarang mendukung OpenSearch versi 1.3. Untuk informasi selengkapnya, lihat catatan rilis 1.3.	27 Juli 2022

Dukungan plugin MLCommons	Amazon OpenSearch Service menambahkan dukungan untuk plugin MLCommons, yang menyediakan serangkaian algoritma pembelajaran mesin umum melalui panggilan transport dan REST API. Anda juga dapat berinteraksi dengan plugin MLCommons melalui perintah PPL.	27 Juli 2022
dukungan volume gp3	Amazon OpenSearch Service menambahkan dukungan untuk tipe volume SSD gp3 EBS General Purpose. Anda dapat menentukan IOPS dan throughput tambahan yang disediakan saat membuat atau memodifikasi domain.	26 Juli 2022
Dokumentasi praktik terbaik yang disempurnakan	Dokumentasi OpenSearch Layanan Amazon menyediakan praktik terbaik operasional yang ditingkatkan dan rekomendasi umum untuk membuat dan mengoperasikan domain OpenSearch Layanan.	6 Juli 2022
Integrasi dengan Service Quotas	Anda sekarang dapat melihat kuota untuk OpenSearch Layanan Amazon, dan meminta peningkatan kuota, dari konsol Service Quotas.	Juni 29, 2022

Kontrol akses berbasis tag untuk API OpenSearch	Anda sekarang dapat menggunakan tag untuk mengontrol akses ke OpenSearch API. Sebelumnya, Anda hanya dapat menggunakan tag untuk mengontrol akses ke API konfigurasi.	16 Juni 2022
Pencarian lintas-cluster di seluruh Wilayah	Pencarian lintas-cluster sekarang didukung Wilayah AWS selama kedua domain menjalankan Elasticsearch versi 7.10 atau yang lebih baru, atau versi apa pun. OpenSearch	14 Juni 2022
Single Kibana 5.6 dukungan	Amazon OpenSearch Service menambahkan dukungan untuk Kibana 5.6.16 tunggal. Dengan Kibana 5.6.16 tunggal, Anda dapat menggunakan Kibana 5.6 sebagai ujung depan Anda saat menghubungkan ke Elasticsearch versi 5.1, 5.3, 5.5, dan 5.6. Anda harus menggunakan perangkat lunak layanan R20220323 atau yang lebih baru untuk menggunakan Kibana 5.6 tunggal.	4 April 2022

R20220323-P1	Amazon OpenSearch Service baru-baru ini merilis pembaruan perangkat lunak layanan R20220323, tetapi pembaruan kemudian dibatalkan karena masalah. Kami menyarankan Anda memperbarui domain Anda ke rilis patch R20220323-P1 atau yang lebih baru, yang memperbaiki masalah.	4 April 2022
OpenSearch 1.2 dukungan	Amazon OpenSearch Service sekarang mendukung OpenSearch versi 1.2. Untuk informasi lebih lanjut, lihat catatan rilis 1.2.	4 April 2022
Observabilitas	Instalasi default OpenSearch Dasbor untuk OpenSearch Layanan Amazon mencakup plugin Observability, yang dapat Anda gunakan untuk memvisualisasikan peristiwa berbasis data menggunakan Piped Processing Language (PPL) untuk menjelajahi dan menanyakan data Anda. Plugin membutuhkan OpenSearch 1.2 atau yang lebih baru dan perangkat lunak layanan R20220323 atau yang lebih baru.	4 April 2022

Dukungan Kibana 7.7.1	Domain Amazon OpenSearch Service yang menjalankan Elasticsearch 7.7 sekarang mendukung rilis patch terbaru untuk Kibana 7.7, yang menambahkan perbaikan bug dan meningkatkan keamanan. Saat Anda memperbarui domain 7.7 Anda ke perangkat lunak layanan R20220323 atau yang lebih baru, OpenSearch Layanan akan secara otomatis memutakhirkannya ke rilis patch ini.	4 April 2022

Perubahan metrik tekanan memori JVM	Amazon OpenSearch Service mengubah logika `JVMMemoryPressure` CloudWatch metrik agar lebih akurat mencerminkan pemanfaatan memori. Sebelumnya, metrik hanya mempertimbangkan kumpulan memori generasi lama dari heap JVM. Dengan perubahan ini, metrik juga mempertimbangkan kumpulan memori generasi muda. Setelah memperbarui domain ke perangkat lunak layanan R20220323, Anda mungkin melihat peningkatan `JVMMemoryPressure`, `MasterJVMMemoryPressure` dan/atau metrik. `WarmJVMMemoryPressure`	4 April 2022
Kamus khusus dengan plugin Analisis IK (Mandarin)	Amazon OpenSearch Service sekarang mendukung penggunaan kamus khusus dengan plugin Analisis IK (Chinese).	4 April 2022

Replikasi lintas-cluster pada domain yang ada	OpenSearch Layanan Amazon menghapus batasan bahwa Anda hanya dapat menerapkan penelusuran lintas klaster dan replikasi lintas klaster pada domain yang dibuat pada atau setelah 3 Juni 2020. Anda sekarang dapat mengaktifkan fitur ini di semua domain terlepas dari kapan mereka dibuat. Kedua domain harus pada perangkat lunak layanan R20220323 atau yang lebih baru.	4 April 2022
Visibilitas penyebaran biru/hijau	Amazon OpenSearch Service sekarang menawarkan lebih banyak visibilitas ke dalam kemajuan penerapan biru/hijau. Anda dapat memantau detail ini di konsol atau menggunakan API konfigurasi.	27 Januari 2022

Kontrol akses berbutir halus pada domain yang ada	Anda sekarang dapat mengaktifkan kontrol akses berbutir halus pada domain yang ada. Anda dapat mengaktifkan periode migrasi sementara untuk kebijakan akses berbasis Open/IP untuk memastikan bahwa pengguna dapat terus mengakses domain Anda saat Anda membuat dan memetakan peran. Mengaktifkan kontrol akses berbutir halus pada domain yang ada memerlukan perangkat lunak layanan R20211203 atau yang lebih baru.	6 Januari 2022
Peran OpenSearch Dasbor Berganti Nama	Dengan perangkat lunak layanan R20211203, `kibana_user` peran tersebut diubah namanya menjadi`opensearch_dashboards_user` , dan diubah namanya menjadi. `kibana_read_only` `opensearch_dashboards_read_only` Perubahan ini berlaku untuk semua yang baru dibuat 1 OpenSearch . x domain. Untuk OpenSearch domain yang ada yang Anda tingkatkan ke perangkat lunak layanan R20211203, perannya tetap sama.	4 Januari 2022

OpenSearch 1.1 dukungan	Amazon OpenSearch Service sekarang mendukung OpenSearch versi 1.1. Untuk informasi selengkapnya, lihat catatan rilis 1.1.	4 Januari 2022
Editor visual ISM	Instalasi default OpenSearch Dasbor untuk OpenSearch Layanan Amazon sekarang mendukung editor visual untuk kebijakan ISM. Fitur ini membutuhkan OpenSearch 1.1 atau yang lebih baru.	4 Januari 2022
Pembaruan pencegahan wakil kebingungan lintas layanan	Amazon OpenSearch Service mendukung penggunaan `aws:SourceArn` dan kunci konteks kondisi `aws:SourceAccount` global dalam kebijakan sumber daya IAM untuk mencegah masalah deputi yang membingungkan. Anda harus menggunakan perangkat lunak layanan R20211203 atau yang lebih baru untuk menggunakan tombol kondisi ini.	4 Januari 2022

Tambalan Log4j	Perangkat lunak layanan R20211203-P2 memperbarui versi Log4j yang digunakan dalam OpenSearch Layanan seperti yang direkomendasikan oleh saran di CVE-2021-44228 dan CVE-2021-45046. Patch berlaku untuk domain yang menjalankan semua versi OpenSearch dan Elasticsearch. OpenSearch Layanan akan terus memperbarui berbagai versi Log4j secara internal, dan mereka tidak akan selalu terbatas pada versi terbaru dari Log4j. Versi Log4j di domain Anda bergantung pada versi perangkat lunak yang dijalankan domain tersebut. Namun, terlepas dari versi Log4j, selama Anda menjalankan R20211203-P2 atau yang lebih baru, domain Anda berisi pembaruan Log4j yang diperlukan untuk menangani CVE-2021-44228 dan CVE-2021-45046.	Desember 15, 2021

Replikasi lintas-cluster	Replikasi lintas klaster memungkinkan Anda mereplikasi indeks, pemetaan, dan metadata dari satu domain Layanan ke domain Layanan lainnya. OpenSearch Replikasi lintas-cluster memerlukan domain yang menjalankan Elasticsearch 7.10 atau 1.1 atau yang lebih baru. OpenSearch	5 Oktober 2021
Kebijakan AWS terkelola baru	Peluncuran OpenSearch Layanan Amazon mencakup kebijakan baru AWS yang dikelola dan penghentian kebijakan lama.	8 September 2021
Dukungan Kibana 6.4.3	Domain Amazon OpenSearch Service yang menjalankan Elasticsearch versi 6.4 lama sekarang mendukung rilis patch terbaru untuk Kibana 6.4, yang menambahkan perbaikan bug dan meningkatkan keamanan. OpenSearch Layanan akan secara otomatis meningkatkan domain ke rilis patch ini.	8 September 2021

Aliran data	Amazon OpenSearch Service menambahkan dukungan untuk aliran data, yang menyederhanakan proses pengelolaan data deret waktu. Domain Anda harus menjalankan OpenSearch 1.0 atau yang lebih baru untuk menggunakan aliran data.	8 September 2021
OpenSearch Layanan Amazon	AWS mengganti nama OpenSearch Layanan Amazon untuk menghapus merek “Elasticsearch” lama. Amazon OpenSearch Service mendukung OpenSearch dan warisan Elasticsearch OSS. Saat Anda membuat cluster, Anda memiliki opsi mesin pencari mana yang akan digunakan. OpenSearch Layanan menawarkan kompatibilitas yang luas dengan Elasticsearch OSS 7.10, versi open source terakhir dari perangkat lunak.	8 September 2021

Penyimpanan dingin	Cold storage adalah tingkat penyimpanan baru untuk data yang jarang diakses atau historis. Indeks dingin hanya menempati penyimpan an S3 dan tidak memiliki komputasi melekat pada mereka. Penyimpanan dingin memerlukan domain yang menjalankan Elasticsearch 7.9 atau yang lebih baru dan perangkat lunak layanan R20210426 atau yang lebih baru.	13 Mei 2021
Instans Graviton berbasis lengan	Amazon OpenSearch Service sekarang mendukung tipe instans Graviton berbasis ARM (M6G, C6G, R6G, dan R6GD). Tipe instans Graviton tersedia di domain baru dan yang sudah ada yang menjalankan Elasticse arch 7.9 atau yang lebih baru dan perangkat lunak layanan R20210331 atau yang lebih baru.	4 Mei 2021

Template ISM	Amazon OpenSearch Service menambahkan dukungan untuk template ISM, yang memungkinkan Anda secara otomatis melampirkan kebijakan ISM ke indeks jika indeks cocok dengan pola yang ditentukan dalam kebijakan. Templat ISM memerlukan layanan perangkat lunak R20210426 atau yang lebih baru. Pembaruan ini juga mengusangkan pengaturan `policy_id` , yang berarti Anda tidak dapat lagi menggunakan templat indeks untuk menerapkan kebijakan ISM untuk indeks yang baru dibuat. Pembaruan memperkenalkan perubahan besar untuk CloudFormation templat yang ada menggunakan pengaturan ini.	27 April 2021
Dukungan Elasticsearch 7.10	Amazon OpenSearch Service sekarang mendukung Elasticsearch versi 7.10. Untuk informasi lebih lanjut, lihat catatan rilis 7.10.	21 April 2021

Pencarian asinkron	Amazon OpenSearch Service sekarang mendukung penelusuran asinkron, yang memungkinkan Anda menjalankan permintaan pencarian di latar belakang. Pencarian asinkron memerlukan domain yang menjalankan Elasticsearch 7.10 atau yang lebih baru dan perangkat lunak layanan R20210331 atau yang lebih baru.	21 April 2021
Kontrol akses berbasis tag untuk API konfigurasi	Anda sekarang dapat menggunakan AWS tag untuk mengontrol akses ke API konfigurasi Amazon ES.	2 Maret 2021
Setel Otomatis	Amazon OpenSearch Service menambahkan Auto-Tune, yang menggunakan metrik kinerja dan penggunaan dari klaster Anda untuk menyarankan perubahan pada setelan JVM di node Anda. Auto-tune memerlukan domain menjalankan Elasticsearch 6.7 atau yang lebih baru dan perangkat lunak layanan R20201117 atau lebih baru.	24 Februari 2021

Analisis Jejak	Instalasi default Kibana untuk OpenSearch Layanan Amazon sekarang mencakup plugin analisis jejak, yang memungkinkan Anda memantau data jejak dari aplikasi terdistribusi Anda. Plugin memerlukan domain yang menjalankan Elasticsearch 7.9 atau yang lebih baru dan perangkat lunak layanan R20210201 atau yang lebih baru.	17 Februari 2021
Metrik pecahan	Amazon OpenSearch Service menambahkan CloudWatch metrik berikut untuk melacak status pecahan:`Shards.active` ,,, `Shards.unassigned` `Shards.delayedUnassigned` `Shards.activePrimary` ,`Shards.initializing` . `Shards.relocating` Metrik tersedia di domain yang menjalankan perangkat lunak layanan R20210201 atau yang lebih baru.	17 Februari 2021

Laporan Kibana	Instalasi default Kibana for Amazon OpenSearch Service sekarang mendukung laporan sesuai permintaan untuk halaman Discover, Visualize , dan Dashboard. Fitur ini memerlukan Elasticsearch 7.9 atau yang lebih baru dan perangkat lunak layanan R20210201 atau yang lebih baru.	17 Februari 2021
Dukungan Kibana 5.6.16	Domain Amazon OpenSearc h Service yang menjalankan Elasticsearch 5.6 sekarang mendukung rilis patch terbaru untuk Kibana 5.6, yang menambahkan perbaikan bug dan meningkatkan keamanan. Amazon ES akan secara otomatis meningkatkan domain ke rilis patch ini.	17 Februari 2021
Enkripsi untuk domain yang ada	Amazon OpenSearch Service sekarang mendukung pengaktifan enkripsi data saat istirahat dan node-to-node enkripsi pada domain yang ada yang menjalankan Elasticsearch 6.7 atau yang lebih baru. Setelah mengaktif kan pengaturan ini, Anda tidak dapat menonaktifkannya.	27 Januari 2021

Indeks ulang jarak jauh	Amazon OpenSearch Service sekarang mendukung pengindeksan ulang jarak jauh, yang memungkinkan Anda memigrasikan indeks dari domain jarak jauh. Fitur ini memerlukan perangkat lunak layanan R20201117 atau yang lebih baru.	24 November 2020
Bahasa Pemrosesan Pipa	Amazon OpenSearch Service sekarang mendukung Piped Processing Language (PPL), bahasa kueri yang memungkinkan Anda menggunakan sintaks pipe (|) untuk menanyakan data yang disimpan di Elasticse arch. Fitur ini memerluka n perangkat lunak layanan R20201117 atau yang lebih baru. Untuk mempelajari selengkapnya, lihat .	24 November 2020
Notebook Kibana	Amazon OpenSearch Service menambahkan dukungan untuk notebook Kibana, yang memungkinkan Anda menggabungkan visualisa si langsung dan teks naratif dalam satu antarmuka. Fitur ini memerlukan perangkat lunak layanan R20201117 atau yang lebih baru.	24 November 2020

Grafik Gantt	Instalasi default Kibana untuk Amazon OpenSearch Service sekarang mendukung jenis visualisasi baru, bagan Gantt. Fitur ini memerluka n perangkat lunak layanan R20201117 atau yang lebih baru.	24 November 2020
Dukungan Elasticsearch 7.9	Amazon OpenSearch Service sekarang mendukung Elasticsearch versi 7.9. Untuk informasi lebih lanjut, lihat 7.9 catatan rilis.	24 November 2020
Pembaruan deteksi anomali	Deteksi anomali untuk Amazon OpenSearch Service menambahkan dukungan untuk kardinalitas tinggi, yang memungkinkan Anda mengkategorikan anomali dengan dimensi seperti alamat IP, ID produk, kode negara, dan sebagainya. Fitur ini memerlukan perangkat lunak layanan R20201117 atau yang lebih baru.	24 November 2020

Pembaruan kamus dinamis	OpenSearch Layanan Amazon sekarang memungkinkan Anda memperbarui penganalisis penelusuran tanpa pengindeksan ulang. Anda dapat memperbarui file kamus pada beberapa atau semua domain Anda, dan Amazon ES melacak versi paket dari waktu ke waktu sehingga Anda memiliki riwayat apa yang berubah dan kapan. Fitur ini memerlukan perangkat lunak layanan R20201019 atau yang lebih baru.	17 November 2020
Titik akhir kustom	OpenSearch Layanan Amazon sekarang mendukung titik akhir kustom, yang memungkinkan Anda memberi domain Amazon ES Anda URL baru. Jika Anda pernah menukar domain, Anda dapat mempertahankan URL yang sama. Fitur ini memerlukan perangkat lunak layanan R20201019 atau yang lebih baru.	5 November 2020

Plugin bahasa baru	Amazon OpenSearch Service sekarang mendukung Analisis IK (China), Analisis Vietnam, dan plugin Analisis Thailand pada domain yang menjalankan Elasticsearch 7.7 atau versi lebih baru dengan perangkat lunak layanan R20201019 atau yang lebih baru.	28 Oktober 2020
Dukungan Elasticsearch 7.8	Amazon OpenSearch Service sekarang mendukung Elasticsearch versi 7.8. Untuk informasi lebih lanjut, lihat 7.8 catatan rilis.	28 Oktober 2020
Otentikasi SAMB untuk Kibana	Amazon OpenSearch Service sekarang mendukung otentikasi SAMP untuk Kibana, yang memungkinkan Anda menggunakan penyedia identitas pihak ketiga untuk masuk ke Kibana, mengelola kontrol akses halus, mencari data Anda, dan membangun visualisasi. Fitur ini memerlukan perangkat lunak layanan R20201019 atau yang lebih baru.	27 Oktober 2020
Contoh T3	Amazon OpenSearch Service sekarang mendukung tipe `t3.small` dan `t3.medium` instance.	23 September 2020

Log audit	Amazon OpenSearch Service sekarang mendukung log audit untuk data Anda, yang memungkinkan Anda melacak upaya login yang gagal, akses pengguna ke indeks, dokumen, dan bidang, dan banyak lagi. Fitur ini memerlukan perangkat lunak layanan R20200910 atau yang lebih baru.	16 September 2020
UltraWarm pemutakhiran	UltraWarm untuk Amazon OpenSearch Service menambahkan metrik baru, setelan baru, antrian migrasi yang lebih besar, dan API pembatalan. Pembaruan ini memerlukan perangkat lunak layanan R20200910 atau yang lebih baru. Untuk informasi selengkapnya, lihat .	14 September 2020
Belajar Rank	Amazon OpenSearch Service sekarang mendukung plugin Learning to Rank open source, yang memungkin kan Anda menggunakan teknologi pembelajaran mesin untuk meningkatkan relevansi penelusuran. Fitur ini memerlukan perangkat lunak layanan R20200721 atau yang lebih baru.	27 Juli 2020

Kesamaan kosinus K-nn	K-Nearest Neighbor (K-nN) sekarang memungkinkan Anda mencari “tetangga terdekat” dengan kesamaan kosinus selain jarak Euclidean . Fitur ini memerlukan perangkat lunak layanan R20200721 atau yang lebih baru.	23 Juli 2020
kompresi gzip	Amazon OpenSearch Service sekarang mendukung kompresi gzip untuk sebagian besar permintaan dan tanggapan HTTP, yang dapat mengurangi latensi dan menghemat bandwidth. Fitur ini memerlukan perangkat lunak layanan R20200721 atau yang lebih baru.	23 Juli 2020
Dukungan Elasticsearch 7.7	Amazon OpenSearch Service sekarang mendukung Elasticsearch versi 7.7. Untuk informasi lebih lanjut, lihat 7.7 catatan rilis.	23 Juli 2020
Layanan peta Kibana	Instalasi default Kibana untuk OpenSearch Layanan Amazon sekarang mencakup server peta WMS, kecuali untuk domain di Wilayah India dan China.	18 Juni 2020

Perbaikan SQL	Dukungan SQL untuk Amazon OpenSearch Service sekarang mendukung banyak operasi baru, antarmuka pengguna Kibana khusus untuk eksplorasi data, dan CLI interaktif. Untuk informasi selengkapnya, lihat .	3 Juni 2020
Pencarian lintas-cluster	Amazon OpenSearch Service memungkinkan Anda melakukan kueri dan agregasi lintas klaster di beberapa domain yang terhubung.	3 Juni 2020
Deteksi anomali	OpenSearch Layanan Amazon memungkinkan Anda mendeteksi anomali secara otomatis dalam waktu hampir nyata.	3 Juni 2020
UltraWarm	UltraWarm penyimpanan untuk Amazon OpenSearch Service telah meninggalkan pratinjau publik dan sekarang tersedia secara umum. Fitur ini sekarang mendukung berbagai versi yang lebih luas dan Wilayah AWS. Untuk informasi selengkapnya, lihat .	5 Mei 2020

Kamus khusus	OpenSearch Layanan Amazon memungkinkan Anda mengunggah file kamus khusus untuk digunakan dengan klaster Anda. File-file ini meningkatkan hasil pencarian Anda dengan memberi tahu Elasticsearch untuk mengabaikan kata-kata berfrekuensi tinggi tertentu atau memperlakukan istilah sebagai setara.	21 April 2020
Elasticsearch 7.4 Support	Amazon OpenSearch Service sekarang mendukung Elasticsearch versi 7.4. Untuk informasi selengkapnya, lihat Versi yang didukung.	12 Maret 2020
K-nn	Amazon OpenSearch Service menambahkan dukungan untuk pencarian K-Nearest Neighbor (k-NN). k-NN memerlukan perangkat lunak layanan R20200302 atau yang lebih baru.	3 Maret 2020

Indeks Manajemen Negara	Amazon OpenSearch Service menambahkan Index State Management (ISM), yang memungkinkan Anda mengotomatiskan tugas rutin, seperti menghapus indeks saat mencapai usia tertentu. Fitur ini memerlukan perangkat lunak layanan R20200302 atau yang lebih baru.	3 Maret 2020
Dukungan Elasticsearch 5.6.16	Amazon OpenSearch Service sekarang mendukung rilis patch terbaru untuk versi 5.6, yang menambahkan perbaikan bug dan meningkatkan keamanan. Amazon ES akan secara otomatis meningkatkan domain 5.6 yang ada ke rilis ini. Perhatikan bahwa rilis Elasticsearch ini salah melaporkan versinya sebagai 5.6.17.	2 Maret 2020
Kontrol akses berbutir halus	Amazon OpenSearch Service sekarang mendukung kontrol akses berbutir halus, yang menawarkan keamanan pada tingkat indeks, dokumen, dan bidang, Kibana multi-tenancy, dan otentikasi dasar HTTP opsional untuk cluster Anda.	11 Februari 2020

UltraWarm penyimpanan (pratinjau)	Amazon OpenSearch Service menambahkan UltraWarm, tingkat penyimpanan hangat baru yang menggunakan Amazon S3 dan solusi caching canggih untuk meningkat kan kinerja. Untuk indeks yang tidak Anda tulis secara aktif dan kueri lebih jarang, UltraWarm penyimpanan menawarkan biaya per GiB yang jauh lebih rendah.	3 Desember 2019
Fitur enkripsi untuk Wilayah China	Enkripsi data saat istirahat dan node-to-node enkripsi sekarang tersedia di Wilayah `cn-north-1` China (Beijing) dan Wilayah `cn-northw est-1` China (Ningxia).	20 November 2019
Memerlukan HTTPS	Anda sekarang dapat mengharuskan semua lalu lintas ke domain Amazon ES Anda tiba melalui HTTPS. Saat mengonfigurasi domain, beri centang kotak Memerluka n HTTPS. Fitur ini memerluka n perangkat lunak layanan R20190808 atau yang lebih baru.	3 Oktober 2019
Dukungan Elasticsearch 7.1 dan 6.8	Amazon OpenSearch Service sekarang mendukung Elasticsearch versi 7.1 dan 6.8. Untuk informasi selengkap nya, lihat Versi yang didukung.	13 Agustus 2019

Cuplikan per jam	Alih-alih snapshot harian, Amazon OpenSearch Service sekarang mengambil snapshot per jam dari domain yang menjalankan Elasticsearch 5.3 dan yang lebih baru sehingga Anda memiliki cadangan yang lebih sering untuk memulihkan data Anda.	8 Juli 2019
Dukungan Elasticsearch 6.7	Amazon OpenSearch Service sekarang mendukung Elasticsearch versi 6.7. Untuk informasi selengkapnya, lihat Versi yang didukung.	29 Mei 2019
Dukungan SQL	Amazon OpenSearch Service sekarang memungkinkan Anda menanyakan data Anda menggunakan SQL. Dukungan SQL memerlukan perangkat lunak layanan R20190418 atau yang lebih baru.	15 Mei 2019
Tipe instans 5-seri	Amazon OpenSearch Service sekarang mendukung jenis instans M5, C5, dan R5. Dibandingkan dengan tipe instans generasi sebelumnya, jenis baru ini menawarkan performa yang lebih baik dengan harga yang lebih rendah. Untuk informasi selengkapnya, lihat Batasan-batasan.	24 April 2019

Dukungan Elasticsearch 6.5	Amazon OpenSearch Service sekarang mendukung Elasticsearch versi 6.5.	8 April 2019
Peringatan	Peringatan untuk OpenSearch Layanan Amazon memberi tahu Anda ketika data dari satu atau beberapa indeks Amazon ES memenuhi persyaratan tertentu. Peringatan membutuhkan perangkat lunak layanan R20190221 atau yang lebih baru.	25 Maret 2019
Dukungan Tiga Availability Zone	OpenSearch Layanan Amazon sekarang mendukung tiga Availability Zone di banyak Wilayah. Rilis ini juga mencakup pengalaman konsol yang efisien. Multi-AZ ini membutuhkan perangkat lunak layanan R20181023 atau yang lebih baru.	7 Februari 2019
Dukungan Elasticsearch 6.4	Amazon OpenSearch Service sekarang mendukung Elasticsearch versi 6.4.	23 Januari 2019
Cluster 200 simpul	Amazon ES sekarang memungkinkan Anda membuat klaster hingga 200 simpul data untuk total 3 PB penyimpanan.	22 Januari 2019

Pembaruan perangkat lunak layanan	Amazon ES sekarang memungkinkan Anda memperbarui perangkat lunak layanan untuk domain Anda secara manual untuk mendapatkan manfaat dari fitur baru dengan lebih cepat atau memperbarui pada waktu lalu lintas yang rendah. Untuk mempelajari selengkapnya, lihat .	20 November 2018
CloudWatch Metrik baru	Amazon ES sekarang menawarkan metrik tingkat-s impul dan Kesehatan klaster baru dan tab Kesehatan instans dalam konsol Amazon ES.	20 November 2018
Dukungan China (Beijing)	OpenSearch Layanan Amazon sekarang tersedia di Wilayah cn-north-1, di mana ia mendukung jenis instans M4, C4, dan R4.	17 Oktober 2018
ode-to-node Enkripsi N	OpenSearch Layanan Amazon sekarang mendukung node-to-node enkripsi, yang membuat data Anda terenkrip si saat Amazon ES mendistri busikannya ke seluruh cluster Anda.	18 September 2018
Upgrade versi di tempat	OpenSearch Layanan Amazon sekarang mendukung peningkatan versi di tempat.	14 Agustus 2018

Dukungan Elasticsearch 6.3 dan 5.6	Amazon OpenSearch Service sekarang mendukung Elasticsearch versi 6.3 dan 5.6.	14 Agustus 2018
Log kesalahan	Amazon ES sekarang memungkinkan Anda mempublikasikan log kesalahan Elasticsearch ke Amazon. CloudWatch	31 Juli 2018
Contoh Cadangan China (Ningxia)	Amazon ES sekarang menawarkan Instans Cadangan di Wilayah China (Ningxia).	29 Mei 2018
Instans Cadangan	Amazon ES sekarang menawarkan dukungan untuk Instans Cadangan.	7 Mei 2018

Pembaruan sebelumnya

Tabel berikut menjelaskan perubahan penting Amazon ES sebelum Mei 2018.

Perubahan	Deskripsi	Tanggal
Autentikasi Amazon Cognito untuk Kibana	Amazon ES sekarang menawarkan perlindungan halaman login untuk Kibana. Untuk mempelajari informasi lebih lanjut, lihat the section called “Otentikasi Amazon Cognito untuk Dasbor OpenSearch ”.	2 April 2018
Support Elasticsearch 6.2	Amazon OpenSearch Service sekarang mendukung Elasticsearch versi 6.2.	14 Maret 2018
Plugin Analisis Korea	Amazon ES sekarang mendukung versi memori dioptimalkan dari plugin analisis Seunjeon bahasa Korea.	13 Maret 2018

Perubahan	Deskripsi	Tanggal
Pembaruan Kontrol Akses Instan	Perubahan kebijakan kontrol akses pada domain Amazon ES sekarang berlaku langsung.	7 Maret 2018
Menskalakan Petabyte	Amazon ES sekarang mendukung tipe instans I3 dan penyimpanan domain total hingga 1,5 PB. Untuk mempelajari informasi lebih lanjut, lihat the section called “Menskalakan Petabyte”.	19 Desember 2017
Enkripsi Data Saat Tidak Digunakan	Amazon ES kini mendukung enkripsi data saat tidak digunakan. Untuk mempelajari informasi lebih lanjut, lihat the section called “Enkripsi diam”.	7 Desember 2017
Mendukung Elasticsearch 6.0	Amazon ES sekarang mendukung Elasticsearch versi 6.0. Untuk petunjuk dan pertimbangan migrasi, lihat the section called “Memutakhirkan domain”.	6 Desember 2017
Dukungan VPC	Amazon ES kini memungkinkan Anda meluncurkan domain di dalam Amazon Virtual Private Cloud. Dukungan VPC menyediakan lapisan tambahan keamanan dan menyederh anakan komunikasi antara Amazon ES dan layanan lainnya dalam VPC. Untuk mempelajari informasi lebih lanjut, lihat the section called “Dukungan VPC”.	17 Oktober 2017
Penerbitan Log Lambat	Amazon ES sekarang mendukung penerbitan log lambat ke CloudWatch Log. Untuk mempelajari informasi lebih lanjut, lihat the section called “Log pemantauan”.	16 Oktober 2017
Mendukung Elasticsearch 5.5	Amazon ES sekarang mendukung Elasticsearch versi 5.5. Anda sekarang dapat memulihkan snapshot otomatis tanpa menghubungi AWS Support dan menyimpan skrip menggunakan API. `_scripts`	7 September 2017
Mendukung Elasticsearch 5.3	Amazon ES menambahkan dukungan untuk Elasticsearch versi 5.3.	1 Juni 2017

Perubahan	Deskripsi	Tanggal
Lebih Banyak Instans dan Kapasitas EBS per Klaster	Amazon ES sekarang mendukung hingga 100 simpul dan kapasitas EBS 150 TB per klaster.	5 April 2017
Mendukung Canada (Central) dan EU (London)	Amazon ES menambahkan dukungan untuk Wilayah berikut: Kanada (Tengah), ca-central-1, dan UE (London), eu-barat-2.	20 Maret 2017
Lebih Banyak Instans dan Volume EBS yang Lebih Besar	Amazon ES menambahkan dukungan untuk lebih banyak instans dan volume EBS yang lebih besar.	21 Februari 2017
Mendukung Elasticsearch 5.1	Amazon ES menambahkan dukungan untuk Elasticsearch versi 5.1.	30 Januari 2017
Mendukung Plugin Analisis Fonetik	Amazon ES sekarang menyediakan integrasi built-in dengan plugin Analisis Fonetik, yang memungkinkan Anda untuk menjalankan kueri “mirip-suara” pada data Anda.	22 Desember 2016
Mendukung US East (Ohio)	Amazon ES menambahkan dukungan untuk Wilayah berikut: AS Timur (Ohio), us-timur-2.	17 Oktober 2016
Metrik Performa Baru	Amazon ES menambahkan metrik performa, `ClusterUsedSpace` .	29 Juli 2016
Mendukung Elasticsearch 2.3	Amazon ES menambahkan dukungan untuk Elasticsearch versi 2.3.	27 Juli 2016
Mendukung Asia Pacific (Mumbai)	Amazon ES menambahkan dukungan untuk Wilayah berikut: Asia Pasifik (Mumbai), ap-selatan-1.	27 Juni 2016
Lebih Banyak Instans per Klaster	Amazon ES meningkatkan jumlah maksimum instans (jumlah instans) per klaster dari 10 sampai 20.	18 Mei 2016

Perubahan	Deskripsi	Tanggal
Mendukung Asia Pacific (Seoul)	Amazon ES menambahkan dukungan untuk Wilayah berikut: Asia Pasifik (Seoul), ap-northeast-2.	28 Januari 2016
Amazon ES	Rilis awal.	1 Oktober 2015

AWSGlosarium

Untuk AWS terminologi terbaru, lihat AWSglosarium di Referensi. Glosarium AWS

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