aws

Panduan Manajemen

Amazon EMR

Amazon EMR: Panduan Manajemen

Table of Contents

Apa itu Amazon EMR?

Amazon EMR (sebelumnya disebut Amazon Elastic MapReduce) adalah platform cluster terkelola yang menyederhanakan menjalankan kerangka kerja data besar, seperti Apache Hadoop dan Apache Spark, untuk memproses dan menganalisis sejumlah besar data. AWS Dengan menggunakan kerangka kerja ini dan proyek sumber terbuka terkait, Anda dapat memproses data untuk tujuan analitik dan beban kerja intelijen bisnis. Amazon EMR juga memungkinkan Anda mengubah dan memindahkan sejumlah besar data ke dalam dan keluar dari penyimpanan data dan database lainnyaAWS, seperti Amazon Simple Storage Service (Amazon S3) Service S3) dan Amazon DynamoDB.

Jika Anda baru pertama kali menggunakan Amazon EMR, sebaiknya Anda memulai dengan membaca berikut ini, sebagai tambahan dari bagian ini:

- Amazon EMR – Halaman layanan ini menyediakan sorotan Amazon EMR, detail produk, dan informasi harga.
- Tutorial: Memulai dengan Amazon EMR – Tutorial ini memungkinkan Anda memulai menggunakan Amazon EMR dengan cepat.

Dalam Bagian Ini

Gambaran Umum Amazon EMR

Topik ini memberikan gambaran umum tentang klaster Amazon EMR, termasuk cara mengirimkan pekerjaan ke klaster, cara data diproses, dan beragam status yang dilewati klaster selama pemrosesan.

Dalam Topik Ini

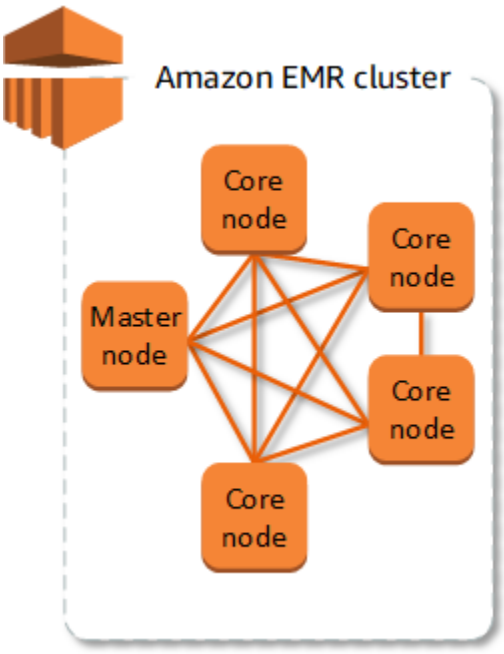
Memahami klaster dan simpul

Komponen sentral dari Amazon EMR adalah klaster. Klaster adalah koleksi instans Amazon Elastic Compute Cloud (Amazon EC2). Setiap instans dalam klaster disebut simpul. Setiap simpul memiliki peran dalam klaster, disebut sebagai jenis simpul. Amazon EMR juga menginstal komponen perangkat lunak yang berbeda pada setiap jenis simpul, memberi setiap simpul peran dalam aplikasi terdistribusi seperti Apache Hadoop.

Jenis simpul di Amazon EMR adalah sebagai berikut:

- Node primer: Node yang mengelola cluster dengan menjalankan komponen perangkat lunak untuk mengoordinasikan distribusi data dan tugas di antara node lain untuk diproses. Node primer melacak status tugas dan memantau kesehatan cluster. Setiap cluster memiliki simpul utama, dan dimungkinkan untuk membuat cluster simpul tunggal hanya dengan simpul utama.
- Simpul Inti: Sebuah simpul dengan komponen perangkat lunak yang menjalankan tugas dan menyimpan data dalam Sistem File Terdistribusi Hadoop (HDFS) pada klaster Anda. Klaster multi-simpul memiliki setidaknya satu simpul inti.
- Simpul tugas: Sebuah simpul dengan komponen perangkat lunak yang hanya menjalankan tugas dan tidak menyimpan data dalam HDFS. Simpul tugas bersifat opsional.

Diagram berikut merupakan cluster dengan satu node utama dan empat node inti.



Mengirim pekerjaan ke sebuah klaster

Ketika Anda menjalankan sebuah klaster di Amazon EMR, Anda memiliki beberapa opsi untuk bagaimana Anda menentukan pekerjaan yang perlu dilakukan.

- Menyediakan seluruh definisi pekerjaan yang harus dilakukan dalam fungsi yang Anda tentukan sebagai langkah-langkah ketika Anda membuat sebuah klaster. Hal ini biasanya dilakukan untuk klaster yang memproses sejumlah set data dan mengakhiri ketika pemrosesan selesai.
- Membuat klaster yang berjalan lama dan menggunakan konsol Amazon EMR, Amazon EMR API, atau AWS CLI untuk mengirimkan langkah, yang mungkin berisi satu atau beberapa pekerjaan. Untuk informasi selengkapnya, lihat Kirim pekerjaan ke sebuah klaster.
- Buat cluster, sambungkan ke node utama dan node lain sesuai kebutuhan menggunakan SSH, dan gunakan antarmuka yang disediakan aplikasi yang diinstal untuk melakukan tugas dan mengirimkan kueri, baik skrip atau interaktif. Untuk informasi selengkapnya, lihat Panduan Rilis Amazon EMR.

Memproses data

Ketika Anda meluncurkan klaster, Anda memilih kerangka kerja dan aplikasi yang akan diinstal untuk kebutuhan pemrosesan data Anda. Untuk memproses data dalam klaster Amazon EMR, Anda dapat mengirimkan pekerjaan atau queri secara langsung ke aplikasi yang diinstal, atau Anda dapat menjalankan langkah dalam klaster.

Mengirimkan pekerjaan secara langsung ke aplikasi

Anda dapat mengirimkan pekerjaan dan berinteraksi langsung dengan perangkat lunak yang diinstal pada klaster Amazon EMR Anda. Untuk melakukan ini, Anda biasanya terhubung ke node utama melalui koneksi aman dan mengakses antarmuka dan alat yang tersedia untuk perangkat lunak yang berjalan langsung di cluster Anda. Untuk informasi selengkapnya, lihat Connect ke sebuah cluster.

Menjalankan langkah-langkah untuk memproses data

Anda dapat mengirimkan satu atau beberapa langkah yang dipesan untuk klaster Amazon EMR. Setiap langkah adalah unit kerja yang berisi instruksi untuk memanipulasi data untuk diproses oleh perangkat lunak yang diinstal pada klaster.

Berikut ini adalah contoh proses menggunakan empat langkah:

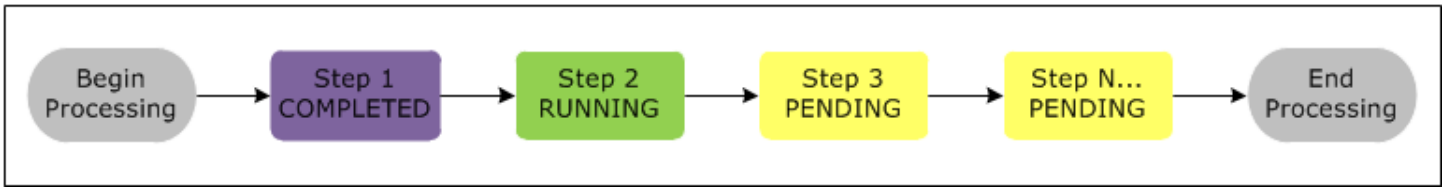
1. Mengirim set data input untuk diproses.
2. Memproses output dari langkah pertama dengan menggunakan program Pig.
3. Memproses set data input kedua dengan menggunakan program Hive.
4. Menulis set data output.

Secara umum, ketika Anda memproses data di Amazon EMR, input adalah data yang disimpan sebagai file dalam sistem file yang mendasari pilihan Anda, seperti Amazon S3 atau HDFS. Data ini melewati dari satu langkah ke langkah berikutnya dalam urutan pemrosesan. Langkah terakhir menulis data output ke lokasi yang ditentukan, seperti bucket Amazon S3.

Langkah dijalankan dalam urutan berikut:

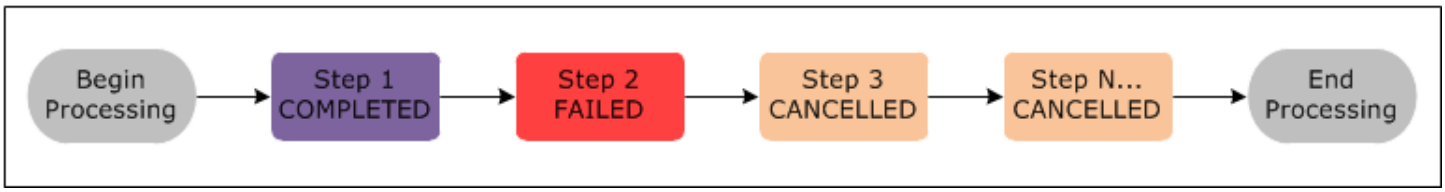
1. Permintaan dikirimkan untuk memulai pemrosesan langkah.
2. Status semua langkah diatur ke PENDING.
3. Ketika langkah pertama dalam urutan dimulai, statusnya berubah menjadi RUNNING. Langkah lainnya tetap dalam status PENDING.
4. Setelah langkah pertama selesai, statusnya berubah menjadi COMPLETED.
5. Langkah selanjutnya dalam urutan dimulai, statusnya berubah menjadi RUNNING. Ketika selesai, status berubah menjadi COMPLETED.
6. Pola ini berulang untuk setiap langkah sampai semuanya selesai dan pemrosesan berakhir.

Diagram berikut merupakan urutan langkah dan perubahan status untuk langkah-langkah saat diproses.



Jika langkah gagal selama pemrosesan, statusnya berubah menjadi FAILED. Anda dapat menentukan apa yang terjadi selanjutnya untuk setiap langkah. Secara default, setiap langkah yang tersisa dalam urutan diatur ke CANCELLED dan tidak berjalan jika langkah sebelumnya gagal. Anda juga dapat memilih untuk mengabaikan kegagalan dan mengizinkan langkah-langkah yang tersisa untuk dilanjutkan, atau untuk mengakhiri klaster segera.

Diagram berikut merupakan urutan langkah dan perubahan default statusnya ketika langkah gagal selama pemrosesan.



Memahami siklus hidup klaster

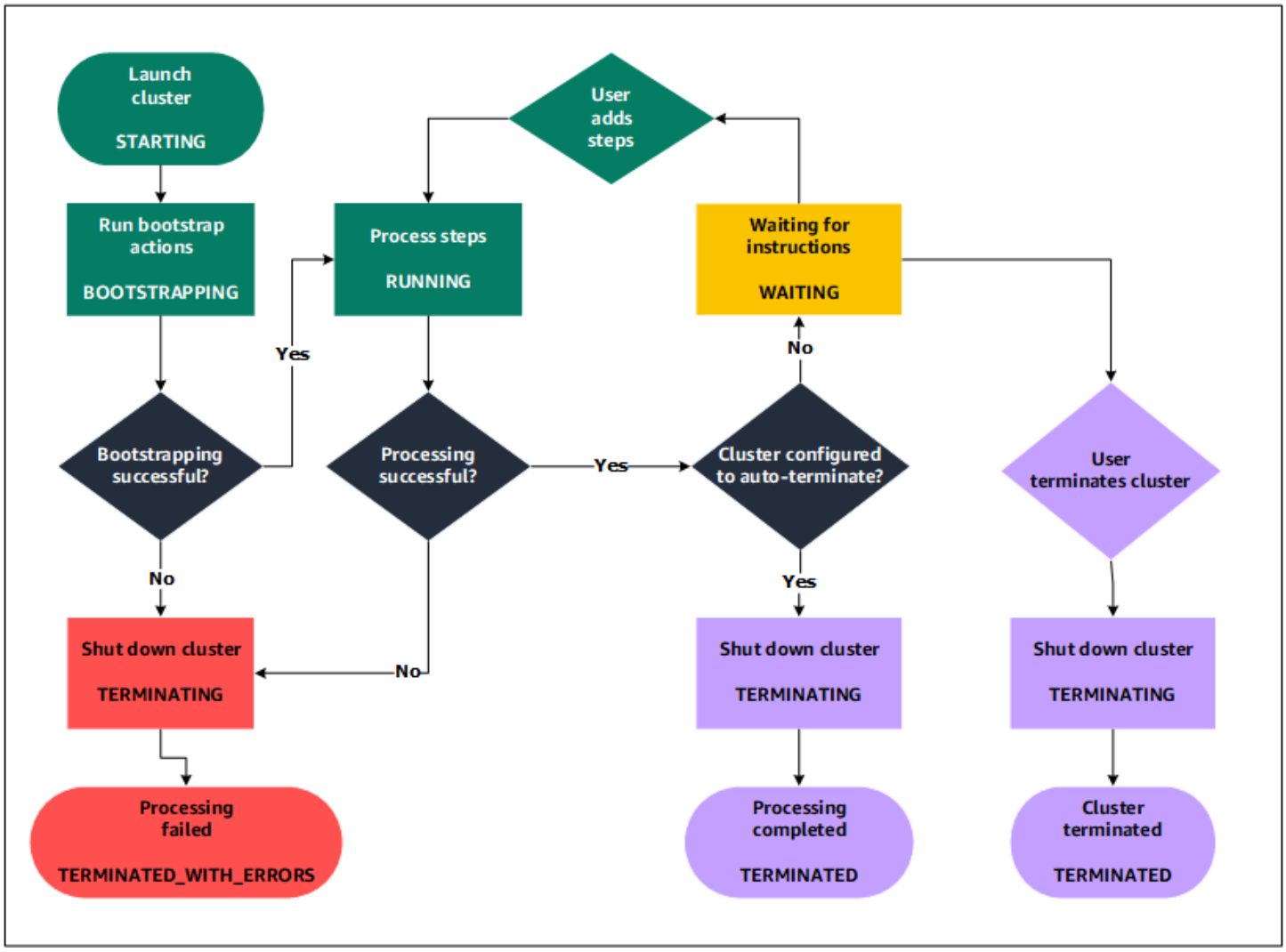
Sebuah klaster Amazon EMR berhasil dengan mengikuti proses ini:

1. Amazon EMR terlebih dahulu menyediakan instans EC2 dalam klaster untuk setiap instans sesuai dengan spesifikasi Anda. Untuk informasi selengkapnya, lihat Konfigurasi perangkat keras dan jaringan klaster. Untuk semua instans, Amazon EMR menggunakan AMI default untuk Amazon EMR atau Amazon Linux AMI khusus yang Anda tentukan. Untuk informasi selengkapnya, lihat Menggunakan AMI kustom. Selama fase ini, status klasternya adalah `STARTING`.
2. Amazon EMR menjalankan tindakan bootstrap yang Anda tentukan pada setiap instans. Anda dapat menggunakan tindakan bootstrap untuk menginstal aplikasi khusus dan melakukan kustomisasi yang Anda perlukan. Untuk informasi selengkapnya, lihat Buat tindakan bootstrap untuk menginstal perangkat lunak tambahan. Selama fase ini, status klasternya adalah `BOOTSTRAPPING`.
3. Amazon EMR menginstal aplikasi native yang Anda tentukan saat membuat klaster, seperti Hive, Hadoop, Spark, dan sebagainya.
4. Setelah tindakan bootstrap berhasil diselesaikan dan aplikasi native diinstal, status klasternya adalah `RUNNING`. Pada titik ini, Anda dapat menyambung ke instans klaster, dan klaster secara berurutan menjalankan langkah-langkah yang telah Anda tentukan ketika membuat klaster. Anda dapat mengirimkan langkah-langkah tambahan, yang berjalan setelah langkah sebelumnya selesai. Untuk informasi selengkapnya, lihat Kirim pekerjaan ke sebuah klaster.
5. Setelah langkah berhasil berjalan, klaster berubah ke status `WAITING`. Jika klaster dikonfigurasi untuk diakhiri otomatis setelah langkah terakhir selesai, klaster berubah ke status `TERMINATING` kemudian ke status `TERMINATED`. Jika klaster dikonfigurasi untuk menunggu, Anda harus secara manual mematikannya ketika Anda tidak lagi membutuhkannya. Setelah Anda secara manual mematikan klaster, itu akan berubah ke status `TERMINATING` kemudian ke status `TERMINATED`.

Kegagalan selama siklus hidup klaster menyebabkan Amazon EMR untuk mengakhiri klaster dan semua instans-nya kecuali Anda mengaktifkan perlindungan penghentian. Jika klaster berakhir karena kegagalan, data yang disimpan pada klaster dihapus, dan status klaster diatur ke `TERMINATED_WITH_ERRORS`. Jika Anda mengaktifkan perlindungan penghentian, Anda dapat

mengambil data dari klaster, kemudian menghapus perlindungan penghentian dan mengakhiri klaster. Untuk informasi selengkapnya, lihat Menggunakan perlindungan pengakhiran.

Diagram berikut merupakan siklus hidup klaster, dan bagaimana setiap tahap siklus hidup memetakan ke status klaster tertentu.



Manfaat menggunakan Amazon EMR

Terdapat banyak manfaat untuk menggunakan Amazon EMR. Bagian ini memberikan gambaran umum manfaat dan tautan ke informasi tambahan untuk membantu Anda menjelajah lebih jauh.

Topik

- Penghematan biaya
- Integrasi AWS

- Deployment
- Skalabilitas dan fleksibilitas
- Keandalan
- Keamanan
- Pemantauan
- Antarmuka manajemen

Penghematan biaya

Harga Amazon EMR bergantung pada jenis instans dan jumlah instans Amazon EC2 yang Anda deploy serta Wilayah tempat Anda meluncurkan klaster. Harga sesuai permintaan menawarkan tarif rendah, tetapi Anda dapat mengurangi biaya lebih jauh dengan membeli Instans Cadangan atau Instans Spot. Instans Spot dapat menawarkan penghematan yang signifikan—lebih rendah sebanyak sepersepuluh dari harga sesuai permintaan dalam beberapa kasus.

> **Note**
>
> Jika Anda menggunakan Amazon S3, Amazon Kinesis, atau DynamoDB dengan klaster EMR Anda, terdapat biaya tambahan untuk layanan tersebut yang ditagih secara terpisah dari penggunaan Amazon EMR Anda.

> **Note**
>
> Saat menyiapkan kluster EMR Amazon di subnet pribadi, sebaiknya Anda juga menyiapkan titik akhir VPC untuk Amazon S3. Jika kluster EMR Anda berada dalam subnet pribadi tanpa titik akhir VPC untuk Amazon S3, Anda akan dikenakan biaya gateway NAT tambahan yang terkait dengan lalu lintas S3 karena lalu lintas antara kluster EMR Anda dan S3 tidak akan tetap berada dalam VPC Anda.

Untuk informasi selengkapnya tentang opsi harga dan detailnya, lihat harga Amazon EMR.

Integrasi AWS

Amazon EMR terintegrasi dengan layanan AWS lainnya untuk menyediakan kemampuan dan fungsionalitas yang terkait dengan jaringan, penyimpanan, keamanan, dan sebagainya, untuk klaster Anda. Daftar berikut memberikan beberapa contoh integrasi ini:

- Amazon EC2 untuk instans yang terdiri atas simpul dalam klaster
- Amazon Virtual Private Cloud (Amazon VPC) untuk mengonfigurasi jaringan virtual tempat Anda meluncurkan instans
- Amazon S3 untuk menyimpan data input dan output
- Amazon CloudWatch untuk memantau kinerja cluster dan mengonfigurasi alarm
- AWS Identity and Access Management (IAM) untuk mengonfigurasi izin
- AWS CloudTrail untuk mengaudit permintaan yang dibuat untuk layanan
- AWS Data Pipeline untuk menjadwalkan dan memulai klaster Anda
- AWS Lake Formation untuk menemukan, membuat katalog, dan mengamankan data di danau data Amazon S3

Deployment

Klaster EMR Anda terdiri dari instans EC2, yang melakukan pekerjaan yang Anda kirimkan ke klaster. Ketika Anda meluncurkan klaster, Amazon EMR mengonfigurasi instans dengan aplikasi yang Anda pilih, seperti Apache Hadoop atau Spark. Pilih ukuran dan jenis instans yang paling sesuai dengan kebutuhan pemrosesan klaster Anda: pemrosesan batch, kueri latensi rendah, data streaming, atau penyimpanan data besar. Untuk informasi selengkapnya tentang tipe instans yang tersedia untuk Amazon EMR, lihat Konfigurasi perangkat keras dan jaringan klaster.

Amazon EMR menawarkan berbagai cara untuk mengonfigurasi perangkat lunak pada klaster Anda. Misalnya, Anda dapat menginstal rilis Amazon EMR dengan satu set aplikasi pilihan yang dapat mencakup kerangka kerja serbaguna, seperti Hadoop, dan aplikasi, seperti Hive, Pig, atau Spark. Anda juga dapat menginstal salah satu dari beberapa distribusi MapR. Amazon EMR menggunakan Amazon Linux, sehingga Anda juga dapat menginstal perangkat lunak pada klaster secara manual menggunakan manajer paket yum atau dari sumbernya. Untuk informasi selengkapnya, lihat Konfigurasi perangkat lunak klaster.

Skalabilitas dan fleksibilitas

Amazon EMR memberikan fleksibilitas untuk menskalakan klaster Anda naik atau turun seiring berubahnya kebutuhan komputasi Anda. Anda dapat mengubah ukuran klaster untuk menambahkan instans untuk beban kerja puncak dan menghapus instans untuk mengontrol biaya ketika beban kerja puncak mereda. Untuk informasi selengkapnya, lihat Secara manual mengubah ukuran klaster berjalan.

Amazon EMR juga menyediakan opsi untuk menjalankan beberapa grup instans sehingga Anda dapat menggunakan Instans Sesuai Permintaan dalam satu grup untuk daya pemrosesan terjamin bersama dengan Instans Spot dalam grup lain agar pekerjaan Anda selesai lebih cepat dan dengan biaya yang lebih rendah. Anda juga dapat mencampur tipe instans yang berbeda untuk mengambil keuntungan dari harga yang lebih baik untuk satu jenis Instans Spot dari yang lain. Untuk informasi selengkapnya, lihat Kapan Anda harus menggunakan Instans Spot?.

Selain itu, Amazon EMR menyediakan fleksibilitas untuk menggunakan beberapa sistem file untuk input, output, dan data menengah. Misalnya, Anda dapat memilih Hadoop Distributed File System (HDFS) yang berjalan pada node primer dan inti klaster Anda untuk memproses data yang tidak perlu Anda simpan di luar siklus hidup klaster Anda. Anda dapat memilih Sistem File EMR (EMRFS) untuk menggunakan Amazon S3 sebagai lapisan data untuk aplikasi yang berjalan di klaster Anda sehingga Anda dapat memisahkan komputasi dan penyimpanan Anda, serta mempertahankan data di luar siklus hidup klaster. EMRFS memberikan manfaat tambahan yang memungkinkan Anda meningkatkan atau mengurangi kebutuhan komputasi dan penyimpanan Anda secara independen. Anda dapat menskalakan kebutuhan komputasi dengan mengubah ukuran klaster dan Anda dapat menskalakan kebutuhan penyimpanan dengan menggunakan Amazon S3. Untuk informasi selengkapnya, lihat Bekerja dengan sistem penyimpanan dan file.

Keandalan

Amazon EMR memantau simpul dalam klaster Anda dan secara otomatis mengakhiri dan mengganti instans apabila mengalami kegagalan.

Amazon EMR menyediakan opsi konfigurasi yang mengontrol jika klaster Anda dihentikan secara otomatis atau manual. Jika Anda mengonfigurasi klaster agar secara otomatis diakhiri, klaster akan diakhiri setelah semua langkah selesai. Ini disebut sebagai klaster sementara. Namun, Anda dapat mengonfigurasi klaster untuk terus berjalan setelah pemrosesan selesai sehingga Anda dapat memilih untuk mengakhirinya secara manual ketika tidak lagi membutuhkannya. Atau, Anda dapat membuat klaster, berinteraksi dengan aplikasi yang diinstal secara langsung, kemudian secara

manual mengakhiri klaster tersebut ketika tidak lagi membutuhkannya. Klaster dalam contoh ini disebut sebagai klaster yang berjalan lama.

Selain itu, Anda dapat mengonfigurasi perlindungan penghentian untuk mencegah instans di klaster Anda diakhiri karena kesalahan atau masalah selama pemrosesan. Ketika perlindungan penghentian diaktifkan, Anda dapat memulihkan data dari instans sebelum penghentian. Pengaturan default untuk opsi ini berbeda bergantung pada apakah Anda memulai klaster menggunakan konsol, CLI, atau API. Untuk informasi selengkapnya, lihat Menggunakan perlindungan pengakhiran.

Keamanan

Amazon EMR memanfaatkan layanan AWS lain, seperti IAM dan Amazon VPC, serta fitur seperti pasangan kunci Amazon EC2, untuk membantu Anda mengamankan klaster dan data Anda.

IAM

Amazon EMR terintegrasi dengan IAM untuk mengelola izin. Anda menentukan izin menggunakan kebijakan IAM, yang Anda lampirkan ke pengguna atau grup IAM. Izin yang Anda tetapkan dalam kebijakan menentukan tindakan yang pengguna atau anggota grup dapat lakukan dan sumber daya yang dapat mereka akses. Untuk informasi selengkapnya, lihat Cara kerja Amazon EMR dengan IAM.

Selain itu, Amazon EMR menggunakan peran IAM untuk layanan Amazon EMR itu sendiri dan profil instans EC2 untuk instans. Peran ini memberikan izin untuk layanan dan instans untuk mengakses layanan AWS atas nama Anda. Terdapat peran default untuk layanan Amazon EMR dan peran default untuk profil instans EC2. Peran default menggunakan kebijakan terkelola AWS, yang dibuat untuk Anda secara otomatis saat pertama kali meluncurkan klaster EMR dari konsol dan memilih izin default. Anda juga dapat membuat IAM role default dari AWS CLI. Jika Anda ingin mengelola izin, bukannya AWS, Anda dapat memilih peran khusus untuk layanan dan profil instans. Untuk informasi selengkapnya, lihat Konfigurasi peran layanan IAM untuk izin Amazon EMR untuk layanan AWS dan sumber daya.

Grup keamanan

Amazon EMR menggunakan grup keamanan untuk mengontrol lalu lintas masuk dan keluar untuk instans EC2 Anda. Saat meluncurkan klaster, Amazon EMR menggunakan grup keamanan untuk instans utama dan grup keamanan untuk dibagikan oleh instans inti/tugas Anda. Amazon EMR mengonfigurasi aturan grup keamanan untuk memastikan komunikasi antara instans dalam klaster.

Secara opsional, Anda dapat mengonfigurasi grup keamanan tambahan dan menetapkannya ke instance utama dan inti/tugas Anda untuk aturan yang lebih maju. Untuk informasi selengkapnya, lihat Mengendalikan lalu lintas jaringan dengan grup keamanan.

Enkripsi

Amazon EMR mendukung enkripsi di sisi klien dan di sisi server Amazon S3 opsional dengan EMRFS untuk membantu melindungi data yang Anda simpan di Amazon S3. Dengan enkripsi di sisi server, Amazon S3 mengenkripsi data Anda setelah mengunggahnya.

Dengan enkripsi di sisi klien, proses enkripsi dan dekripsi terjadi di klien EMRFS di klaster EMR Anda. Anda mengelola kunci root untuk enkripsi sisi klien menggunakan AWS Key Management Service (AWS KMS) atau sistem manajemen kunci Anda sendiri.

Untuk informasi selengkapnya, lihat Menentukan enkripsi Amazon S3 menggunakan properti EMRFS.

Amazon VPC

Amazon EMR mendukung peluncuran klaster dalam virtual private cloud (VPC) di Amazon VPC. VPC adalah jaringan virtual terisolasi di AWS yang menyediakan kemampuan untuk mengontrol aspek lanjutan dari konfigurasi jaringan dan akses. Untuk informasi selengkapnya, lihat Mengkonfigurasi jaringan.

AWS CloudTrail

Amazon EMR terintegrasi dengan CloudTrail untuk mencatat informasi tentang permintaan yang dibuat oleh atau atas nama akun Anda. AWS Dengan informasi ini, Anda dapat melacak siapa yang mengakses klaster Anda dan kapan, dan alamat IP asal mereka membuat permintaan. Untuk informasi selengkapnya, lihat Logging panggilan API Amazon EMR di AWS CloudTrail.

Pasangan kunci Amazon EC2

Anda dapat memantau dan berinteraksi dengan cluster Anda dengan membentuk koneksi aman antara komputer jarak jauh Anda dan node utama. Anda menggunakan protokol jaringan Secure Shell (SSH) untuk koneksi ini atau menggunakan Kerberos untuk autentikasi. Jika Anda menggunakan SSH, pasangan kunci Amazon EC2 diperlukan. Untuk informasi selengkapnya, lihat Menggunakan key pair EC2 untuk kredensi SSH.

Pemantauan

Anda dapat menggunakan file log dan antarmuka manajemen Amazon EMR untuk memecahkan masalah klaster, seperti kegagalan atau kesalahan. Amazon EMR menyediakan kemampuan untuk mengarsipkan file log di Amazon S3 sehingga Anda dapat menyimpan log dan memecahkan masalah bahkan setelah klaster Anda berakhir. Amazon EMR juga menyediakan alat debugging opsional di konsol Amazon EMR untuk menelusuri file log berdasarkan langkah, pekerjaan, dan tugas. Untuk informasi selengkapnya, lihat Konfigurasi pencatatan log dan debugging klaster.

Amazon EMR terintegrasi dengan CloudWatch untuk melacak metrik kinerja untuk klaster dan pekerjaan di dalam klaster. Anda dapat mengonfigurasi alarm berdasarkan berbagai metrik, seperti apakah klaster dalam keadaan diam atau persentase penyimpanan yang digunakan. Untuk informasi selengkapnya, lihat Memantau metrik Amazon EMR dengan CloudWatch.

Antarmuka manajemen

Ada beberapa cara berinteraksi dengan Amazon EMR:

- Konsol — Antarmuka pengguna grafis yang dapat Anda gunakan untuk meluncurkan dan mengelola klaster. Dengan itu, Anda mengisi formulir web untuk menentukan detail klaster untuk memulai, melihat detail klaster yang ada, men-debug, dan mengakhiri klaster. Menggunakan konsol adalah cara paling mudah untuk memulai Amazon EMR; tidak memerlukan pengetahuan pemrograman. Konsol tersedia online di https://console.aws.amazon.com/elasticmapreduce/home.
- AWS Command Line Interface (AWS CLI) — Sebuah aplikasi klien yang Anda jalankan pada mesin lokal untuk terhubung ke Amazon EMR serta membuat dan mengelola klaster. AWS CLI berisi serangkaian perintah kaya fitur khusus untuk Amazon EMR. Dengan itu, Anda dapat menulis skrip yang mengotomatiskan proses peluncuran dan pengelolaan klaster. Jika Anda lebih suka bekerja dari baris perintah, menggunakan AWS CLI adalah opsi terbaik. Untuk informasi lebih lanjut, lihat Amazon EMR dalam Referensi Perintah AWS CLI.
- Kit Pengembangan Perangkat Lunak (SDK) — SDK menyediakan fungsi yang memanggil Amazon EMR untuk membuat dan mengelola klaster. Dengan SDK, Anda dapat menulis aplikasi yang mengotomatiskan proses pembuatan dan pengelolaan klaster. Menggunakan SDK adalah opsi terbaik untuk memperluas atau menyesuaikan fungsi Amazon EMR. Amazon EMR saat ini tersedia dalam SDK berikut: Go, Java, NET (C# dan VB.NET), Node.js, PHP, Python, dan Ruby. Untuk informasi selengkapnya tentang SDK ini, lihat Alat untuk AWS dan kode sampel & pustaka Amazon EMR.

- Layanan Web API — Antarmuka tingkat rendah yang dapat Anda gunakan untuk memanggil layanan web secara langsung, menggunakan JSON. Menggunakan API ini adalah opsi terbaik untuk membuat SDK khusus yang memanggil Amazon EMR. Untuk informasi lebih lanjut, lihat Referensi Amazon EMR API.

Gambaran umum arsitektur Amazon EMR

Arsitektur layanan Amazon EMR terdiri dari beberapa lapisan, yang masing-masing menyediakan kemampuan dan fungsi tertentu untuk klaster. Bagian ini memberikan gambaran umum tentang lapisan dan komponen masing-masing.

Dalam Topik Ini

Penyimpanan

Lapisan penyimpanan mencakup sistem file yang berbeda yang digunakan dengan klaster Anda. Terdapat beberapa jenis opsi penyimpanan sebagai berikut.

Sistem File Terdistribusi Hadoop (HDFS)

Sistem File Terdistribusi Hadoop (HDFS) adalah sistem file terdistribusi dan dapat diskalakan untuk Hadoop. HDFS mendistribusikan data yang disimpan di seluruh instans di klaster, menyimpan beberapa salinan data pada instans yang berbeda untuk memastikan tidak ada data yang hilang jika instans individu gagal. HDFS adalah penyimpanan sementara yang diklaim ulang ketika Anda mengakhiri sebuah klaster. HDFS berguna untuk caching hasil antara selama MapReduce pemrosesan atau untuk beban kerja yang memiliki I/O acak yang signifikan.

Untuk informasi lebih lanjut, lihat Penyimpanan instans di panduan ini atau kunjungi Panduan Pengguna HDFS di situs web Apache Hadoop.

EMR File System (EMRFS)

Dengan menggunakan EMR File System (EMRFS), Amazon EMR memperluas Hadoop untuk menambahkan kemampuan untuk mengakses data secara langsung yang tersimpan di Amazon S3 seolah-olah itu adalah sistem file seperti HDFS. Anda dapat menggunakan HDFS atau Amazon S3 sebagai sistem file dalam klaster Anda. Paling sering, Amazon S3 digunakan untuk menyimpan data input dan output dan hasil intermediate yang disimpan dalam HDFS.

Sistem file lokal

Sistem file lokal mengacu pada disk yang terhubung secara lokal. Ketika Anda membuat klaster Hadoop, setiap simpul dibuat dari instans Amazon EC2 yang datang dengan blok yang telah dikonfigurasi dari penyimpanan disk yang telah terlampir yang disebut penyimpanan instans. Data pada volume penyimpanan instans hanya bertahan selama masa hidup instans Amazon EC2-nya.

Manajemen sumber daya klaster

Lapisan manajemen sumber daya bertanggung jawab untuk mengelola sumber daya klaster dan menjadwalkan pekerjaan untuk memproses data.

Secara default, Amazon EMR menggunakan YARN (Yet Another Resource Negotiator), yang merupakan komponen yang diperkenalkan di Apache Hadoop 2.0 untuk mengelola sumber daya klaster secara terpusat untuk beberapa kerangka kerja pemrosesan data. Namun, terdapat kerangka kerja dan aplikasi lain yang ditawarkan di Amazon EMR yang tidak menggunakan YARN sebagai manajer sumber daya. Amazon EMR juga memiliki agen pada setiap simpul yang mengelola komponen YARN, menjaga klaster tetap sehat, dan berkomunikasi dengan Amazon EMR.

Karena Instans Spot sering digunakan untuk menjalankan simpul tugas, Amazon EMR memiliki fungsi default untuk penjadwalan pekerjana YARN sehingga menjalankan pekerjaan tidak akan gagal ketika simpul tugas yang berjalan di Instans Spot diakhiri. Amazon EMR melakukan ini dengan mengizinkan proses utama aplikasi berjalan hanya pada simpul inti. Proses utama aplikasi mengontrol tugas yang sedang berjalan dan harus tetap hidup selama masa tugas.

Amazon EMR merilis 5.19.0 dan yang lebih baru menggunakan fitur label node YARN bawaan untuk mencapai ini. (Versi sebelumnya menggunakan patch kode). Properti dalam klasifikasi konfigurasi `yarn-site` dan `capacity-scheduler` dikonfigurasi secara default sehingga YARN capacity-scheduler dan fair-scheduler memanfaatkan label simpul. Amazon EMR secara otomatis melabeli simpul inti dengan label `CORE`, dan menetapkan properti sehingga utama aplikasi dijadwalkan hanya

pada simpul dengan label INTI. Secara manual memodifikasi properti terkait di klasifikasi konfigurasi yarn-site dan penjadwal kapasitas, atau secara langsung dalam file XML terkait, dapat merusak fitur ini atau memodifikasi fungsi ini.

Kerangka kerja pemrosesan data

Lapisan kerangka kerja pemrosesan data adalah mesin yang digunakan untuk memproses dan menganalisis data. Terdapat banyak kerangka kerja yang tersedia yang berjalan pada YARN atau memiliki manajemen sumber daya mereka sendiri. Kerangka kerja yang berbeda tersedia untuk berbagai jenis kebutuhan pemrosesan, seperti batch, interaktif, dalam memori, streaming, dan sebagainya. Kerangka kerja yang Anda pilih bergantung pada kasus penggunaan Anda. Ini memberi dampak pada bahasa dan antarmuka yang tersedia dari lapisan aplikasi, yang merupakan lapisan yang digunakan untuk berinteraksi dengan data yang ingin Anda proses. Kerangka kerja pemrosesan utama yang tersedia untuk Amazon EMR adalah MapReduce Hadoop dan Spark.

Hadoop MapReduce

Hadoop MapReduce adalah model pemrograman open-source untuk komputasi terdistribusi. Alat ini menyederhanakan proses penulisan aplikasi terdistribusi paralel dengan menangani semua logika, sementara Anda memberikan fungsi Map dan Reduce. Fungsi Map memetakan data untuk mengatur pasangan nilai kunci yang disebut hasil intermediate. Fungsi Reduce menggabungkan hasil intermediate, menerapkan algoritme tambahan, dan memproduksi output akhir. Ada beberapa kerangka kerja yang tersedia untuk MapReduce, seperti Hive, yang secara otomatis menghasilkan program Map dan Reduce.

Untuk informasi lebih lanjut, buka Bagaimana operasi map dan reduce sebenarnya dilakukan di situs web Apache Hadoop Wiki.

Apache Spark

Spark adalah kerangka kerja klaster dan model pemrograman untuk memproses beban kerja big data. Seperti Hadoop MapReduce, Spark adalah sistem pemrosesan terdistribusi open-source tetapi menggunakan grafik asiklik terarah untuk rencana eksekusi dan caching dalam memori untuk kumpulan data. Ketika Anda menjalankan Spark di Amazon EMR, Anda dapat menggunakan EMRFS untuk secara langsung mengakses data Anda di Amazon S3. Spark mendukung beberapa modul kueri interaktif seperti SparkSQL.

Untuk informasi selengkapnya, lihat Apache Spark pada klaster Amazon EMR di Panduan Rilis Amazon EMR.

Aplikasi dan program

Amazon EMR mendukung banyak aplikasi seperti Hive, Pig, dan perpustakaan Spark Streaming untuk menyediakan kemampuan seperti menggunakan bahasa tingkat yang lebih tinggi untuk membuat beban kerja pemrosesan, memanfaatkan algoritme pembelajaran mesin, membuat aplikasi pemrosesan aliran, dan membangun gudang data. Selain itu, Amazon EMR juga mendukung proyek sumber terbuka yang memiliki fungsi manajemen klaster mereka sendiri daripada menggunakan YARN.

Anda menggunakan berbagai pustaka dan bahasa untuk berinteraksi dengan aplikasi yang Anda jalankan di Amazon EMR. Misalnya, Anda dapat menggunakan Java, Hive, atau Pig dengan MapReduce atau Spark Streaming, Spark SQL, MLLib, dan GraphX dengan Spark.

Untuk informasi selengkapnya, lihat Panduan Rilis Amazon EMR.

Mengatur Amazon EMR

Selesaikan tugas dalam bagian ini sebelum meluncurkan klaster Amazon EMR untuk pertama kalinya:

Sebelum Anda menggunakan Amazon EMR untuk pertama kalinya, selesaikan tugas-tugas berikut:

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 mendaftar Akun AWS, 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

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

  Untuk petunjuk, lihat Memulai dalam Panduan Pengguna AWS IAM Identity Center.

Masuk sebagai pengguna administratif

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

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

Buat pasangan kunci Amazon EC2 untuk SSH

> **Note**
>
> Dengan Amazon EMR rilis versi 5.10.0 atau yang lebih baru, Anda dapat mengonfigurasi Kerberos untuk mengautentikasi pengguna dan koneksi SSH ke klaster. Untuk informasi selengkapnya, lihat Gunakan Kerberos untuk otentikasi dengan Amazon EMR.

Untuk mengautentikasi dan terhubung ke node dalam suatu klaster melalui saluran aman menggunakan protokol Secure Shell (SSH), buat pasangan kunci Amazon Elastic Compute Cloud (Amazon EC2) sebelum Anda meluncurkan klaster. Anda juga dapat membuat klaster tanpa pasangan kunci. Hal ini biasanya dilakukan dengan klaster sementara yang memulai, menjalankan langkah-langkah, dan mengakhiri secara otomatis.

Jika...	Maka...
Anda sudah memiliki pasangan kunci Amazon EC2 yang ingin Anda gunakan, atau Anda tidak perlu mengautentikasi ke klaster Anda.	Lewati langkah ini.
Anda perlu membuat pasangan kunci.	Lihat Membuat pasangan kunci Anda menggunakan Amazon EC2.

Langkah selanjutnya

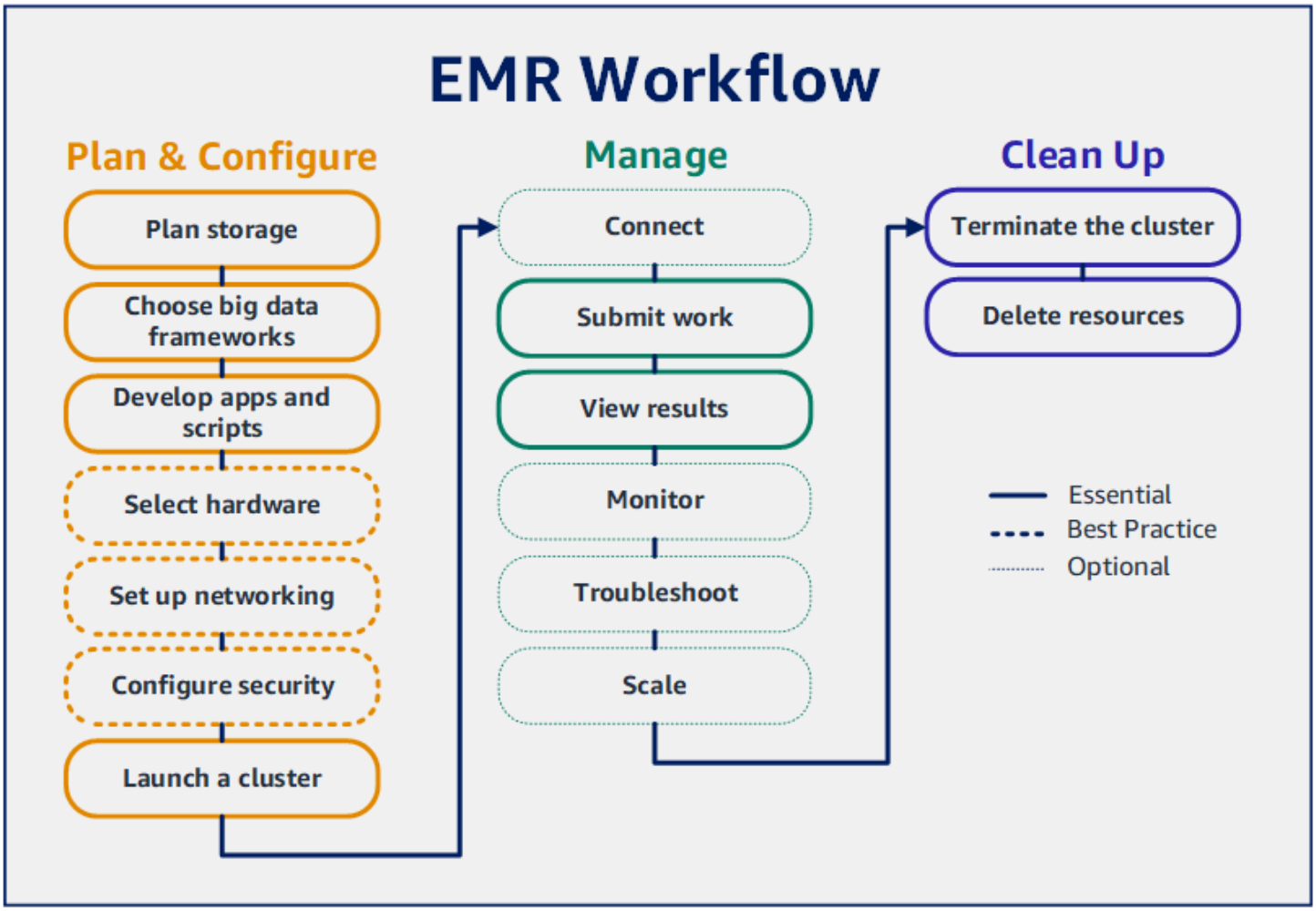
- Untuk panduan tentang membuat klaster sampel, lihat Tutorial: Memulai dengan Amazon EMR.
- Untuk informasi lebih lanjut tentang cara mengonfigurasi klaster khusus dan akses kontrol ke sana, lihat Merencanakan dan mengonfigurasi klaster dan Keamanan di Amazon EMR.

Tutorial: Memulai dengan Amazon EMR

Gambaran Umum

Dengan Amazon EMR, Anda dapat menyiapkan klaster untuk memproses dan menganalisis data dengan kerangka kerja big data hanya dalam beberapa menit. Tutorial ini menunjukkan cara meluncurkan cluster sampel menggunakan Spark, dan cara menjalankan PySpark skrip sederhana yang disimpan dalam bucket Amazon S3. Tutorial ini membahas tugas-tugas penting Amazon EMR dalam tiga kategori alur kerja utama: Rencanakan dan Konfigurasi, Kelola, dan Bersihkan.

Anda akan menemukan tautan ke topik yang lebih rinci saat Anda mengerjakan tutorial, dan ide untuk langkah-langkah tambahan di Langkah selanjutnya bagian ini. Jika Anda memiliki pertanyaan atau bingung, hubungi tim Amazon EMR di Forum diskusi kami.



Prasyarat

- Sebelum Anda meluncurkan klaster Amazon EMR, pastikan Anda menyelesaikan tugas dalam Mengatur Amazon EMR.

Biaya

- Klaster sampel yang Anda buat berjalan di lingkungan langsung. Cluster dikenakan biaya minimal. Untuk menghindari biaya tambahan, pastikan Anda menyelesaikan tugas pembersihan di langkah terakhir dari tutorial ini. Biaya bertambah pada tarif per detik sesuai dengan harga Amazon EMR. Biaya juga bervariasi menurut Wilayah. Untuk informasi lebih lanjut, lihat Harga Amazon EMR.
- Biaya minimal mungkin timbul untuk file kecil yang Anda simpan di Amazon S3. Beberapa atau semua biaya untuk Amazon S3 mungkin dibebaskan jika Anda berada dalam batas penggunaan Tingkat AWS Gratis. Untuk informasi selengkapnya, lihat Harga Amazon S3 dan Tingkat Gratis AWS.

Langkah 1: Rencanakan dan konfigurasi klaster Amazon EMR

Menyiapkan penyimpanan untuk Amazon EMR

Saat Anda menggunakan Amazon EMR, Anda dapat memilih dari berbagai sistem file untuk menyimpan data input, data keluaran, dan file log. Dalam tutorial ini, Anda menggunakan EMRFS untuk menyimpan data dalam bucket S3. EMRFS adalah implementasi dari sistem file Hadoop yang memungkinkan Anda membaca dan menulis file biasa ke Amazon S3. Untuk informasi selengkapnya, lihat Bekerja dengan sistem penyimpanan dan file.

Untuk membuat bucket untuk tutorial ini, ikuti petunjuk di Bagaimana cara membuat bucket S3? dalam Panduan Pengguna Amazon Simple Storage Service Console. Buat bucket dalam Wilayah AWS yang sama tempat Anda berencana meluncurkan klaster Amazon EMR. Misalnya, US West (Oregon) us-west-2.

Bucket dan folder yang Anda gunakan dengan Amazon EMR memiliki keterbatasan berikut:

- Nama dapat terdiri dari huruf kecil, angka, titik (.), dan tanda hubung (-).
- Nama tidak dapat diakhiri dengan angka.
- Nama bucket harus unik di seluruh akun AWS.
- Folder output harus kosong.

Menyiapkan aplikasi dengan data input untuk Amazon EMR

Cara paling umum untuk menyiapkan aplikasi untuk Amazon EMR adalah dengan mengunggah aplikasi dan data inputnya ke Amazon S3. Kemudian, ketika Anda mengirimkan pekerjaan ke klaster Anda, Anda menentukan lokasi Amazon S3 untuk skrip dan data Anda.

Pada langkah ini, Anda mengunggah PySpark skrip sampel ke bucket Amazon S3 Anda. Kami telah menyediakan PySpark skrip untuk Anda gunakan. Skrip memproses data inspeksi pembentukan makanan dan mengembalikan file hasil di bucket S3 Anda. File hasil mencantumkan sepuluh perusahaan teratas dengan pelanggaran jenis “Merah” paling banyak.

Anda juga mengunggah data input sampel ke Amazon S3 agar PySpark skrip dapat diproses. Data input adalah versi modifikasi dari hasil inspeksi Departemen Kesehatan di King County, Washington, dari 2006 hingga 2020. Untuk informasi selengkapnya, lihat King County Open Data: Food Establishment Inspection Data. Berikut ini adalah baris sampel dari set data.

```
name, inspection_result, inspection_closed_business, violation_type, violation_points
100 LB CLAM, Unsatisfactory, FALSE, BLUE, 5
100 PERCENT NUTRICION, Unsatisfactory, FALSE, BLUE, 5
7-ELEVEN #2361-39423A, Complete, FALSE, , 0
```

Untuk mempersiapkan contoh PySpark script untuk ESDM

1. Salin contoh kode di bawah ini ke file baru di editor pilihan Anda.

```
import argparse

from pyspark.sql import SparkSession

def calculate_red_violations(data_source, output_uri):
    """
    Processes sample food establishment inspection data and queries the data to
 find the top 10 establishments
    with the most Red violations from 2006 to 2020.

    :param data_source: The URI of your food establishment data CSV, such as 's3://
DOC-EXAMPLE-BUCKET/food-establishment-data.csv'.
    :param output_uri: The URI where output is written, such as 's3://DOC-EXAMPLE-
BUCKET/restaurant_violation_results'.
    """
```

```
    with SparkSession.builder.appName("Calculate Red Health
 Violations").getOrCreate() as spark:
        # Load the restaurant violation CSV data
        if data_source is not None:
            restaurants_df = spark.read.option("header", "true").csv(data_source)

        # Create an in-memory DataFrame to query
        restaurants_df.createOrReplaceTempView("restaurant_violations")

        # Create a DataFrame of the top 10 restaurants with the most Red violations
        top_red_violation_restaurants = spark.sql("""SELECT name, count(*) AS
 total_red_violations
          FROM restaurant_violations
          WHERE violation_type = 'RED'
          GROUP BY name
          ORDER BY total_red_violations DESC LIMIT 10""")

        # Write the results to the specified output URI
        top_red_violation_restaurants.write.option("header",
 "true").mode("overwrite").csv(output_uri)

if __name__ == "__main__":
    parser = argparse.ArgumentParser()
    parser.add_argument(
        '--data_source', help="The URI for you CSV restaurant data, like an S3
 bucket location.")
    parser.add_argument(
        '--output_uri', help="The URI where output is saved, like an S3 bucket
 location.")
    args = parser.parse_args()

    calculate_red_violations(args.data_source, args.output_uri)
```

2. Simpan file sebagai `health_violations.py`.
3. Unggah `health_violations.py` ke Amazon S3 ke bucket yang Anda buat untuk tutorial ini. Untuk petunjuknya, lihat Mengunggah objek ke bucket di Panduan Memulai Layanan Penyimpanan Sederhana Amazon.

Untuk menyiapkan data input sampel untuk EMR

1. Unduh file zip, food_establishment_data.zip.

2. Unzip dan simpan `food_establishment_data.zip` seperti `food_establishment_data.csv` pada mesin Anda.
3. Unggah file CSV ke bucket S3 yang telah Anda buat untuk tutorial ini. Untuk petunjuknya, lihat Mengunggah objek ke bucket di Panduan Memulai Layanan Penyimpanan Sederhana Amazon.

Untuk informasi lebih lanjut tentang penyiapan data untuk EMR, lihat Mempersiapkan data input.

Meluncurkan klaster Amazon EMR

Setelah menyiapkan lokasi penyimpanan dan aplikasi, Anda dapat meluncurkan sampel klaster Amazon EMR. Pada langkah ini, Anda meluncurkan klaster Apache Spark menggunakan versi rilis Amazon EMR terbaru.

New console

Untuk meluncurkan cluster dengan Spark diinstal dengan konsol baru

1. Masuk keAWS Management Console, dan buka konsol Amazon EMR di https://console.aws.amazon.com/emr.
2. Di bawah EMR pada EC2 di panel navigasi kiri, pilih Cluster, dan kemudian pilih Create cluster.
3. Pada halaman Create Cluster, perhatikan nilai default untuk Release, Instance type, Number of instance, dan Permissions. Bidang ini secara otomatis diisi dengan nilai-nilai yang bekerja untuk klaster tujuan umum.
4. Di bidang Nama klaster, masukkan nama klaster unik untuk membantu Anda mengidentifikasi klaster Anda, seperti klaster pertama *saya*.
5. Di bawah Aplikasi, pilih opsi Spark untuk menginstal Spark pada klaster Anda.

> **Note**
>
> Pilih aplikasi yang Anda inginkan di klaster Amazon EMR sebelum meluncurkan klaster. Anda tidak dapat menambah atau menghapus aplikasi dari klaster setelah peluncuran.

6. Di bawah Log klaster, pilih kotak centang Publikasikan log khusus klaster ke Amazon S3. Ganti nilai lokasi Amazon S3 dengan bucket Amazon S3 yang Anda buat, diikuti oleh. **`/logs`** Sebagai contoh, **`s3://DOC-EXAMPLE-BUCKET/logs`**. Menambahkan **`/logs`** membuat

folder baru yang disebut 'log' di bucket Anda, tempat Amazon EMR dapat menyalin file log klaster Anda.

7. Di bawah Konfigurasi dan izin keamanan, pilih pasangan kunci EC2 Anda. Di bagian yang sama, pilih menu tarik-turun peran Layanan untuk Amazon EMR dan pilih EMR_. DefaultRole Kemudian, pilih peran IAM untuk menu dropdown profil misalnya dan pilih EMR_EC2_. DefaultRole
8. Pilih Create cluster untuk meluncurkan klaster dan membuka halaman detail klaster.
9. Temukan Status klaster di sebelah nama klaster. Status berubah dari Mulai Menjalankan ke Menunggu karena Amazon EMR menyediakan klaster. Anda mungkin perlu memilih ikon refresh di sebelah kanan atau menyegarkan browser Anda untuk melihat pembaruan status.

Status klaster Anda berubah menjadi Menunggu saat klaster aktif, berjalan, dan siap menerima pekerjaan. Untuk informasi selengkapnya tentang membaca ringkasan klaster, lihat Melihat status dan detail klaster. Untuk informasi tentang status klaster, lihat Memahami siklus hidup klaster.

Old console

Untuk meluncurkan cluster dengan Spark diinstal dengan konsol lama

1. Arahkan ke konsol Amazon EMR baru dan pilih Beralih ke konsol lama dari navigasi samping. Untuk informasi selengkapnya tentang apa yang diharapkan saat Anda beralih ke konsol lama, lihat Menggunakan konsol lama.
2. Pilih Buat klaster untuk membuka Wizard Opsi Cepat.
3. Perhatikan nilai default untuk Rilis, Jenis Instans, Jumlah instance, dan Izin pada halaman Buat Klaster - Opsi Cepat. Bidang ini mengisi otomatis dengan nilai-nilai yang bekerja untuk klaster tujuan umum.
4. Masukkan Nama klaster untuk membantu Anda mengidentifikasi klaster. Misalnya, *`Cluster pertama saya`*.
5. Biarkan Logging diaktifkan, tetapi ganti nilai folder S3 dengan bucket Amazon S3 yang Anda buat, diikuti oleh **`/logs`**. Misalnya, **`s3://DOC-EXAMPLE-BUCKET/logs`**. Menambahkan **`/logs`** membuat folder baru yang disebut 'log' di bucket Anda, di mana EMR dapat menyalin file log klaster Anda.
6. Pilih opsi Spark di bawah Aplikasi untuk menginstal Spark di klaster Anda.

> **Note**
>
> Pilih aplikasi yang Anda inginkan di klaster Amazon EMR sebelum meluncurkan klaster. Anda tidak dapat menambah atau menghapus aplikasi dari klaster setelah peluncuran.

7. Pilih pasangan kunci EC2 Anda di bawah Keamanan dan akses.
8. Pilih Buat klaster untuk meluncurkan klaster dan membuka halaman status klaster.
9. Temukan Status klaster di sebelah nama klaster. Status berubah dari Mulai Menjalankan ke Menunggu karena Amazon EMR menyediakan klaster. Anda mungkin perlu memilih ikon refresh di sebelah kanan atau menyegarkan browser Anda untuk melihat pembaruan status.

Status klaster Anda berubah menjadi Menunggu saat klaster aktif, berjalan, dan siap menerima pekerjaan. Untuk informasi selengkapnya tentang membaca ringkasan klaster, lihat Melihat status dan detail klaster. Untuk informasi tentang status klaster, lihat Memahami siklus hidup klaster.

CLI

Untuk meluncurkan cluster dengan Spark diinstal dengan AWS CLI

1. Buat peran default IAM yang kemudian dapat Anda gunakan untuk membuat klaster Anda dengan menggunakan perintah berikut.

```
aws emr create-default-roles
```

Untuk informasi selengkapnya`create-default-roles`, lihat Referensi AWS CLI Perintah.

2. Buat klaster Spark Anda dengan perintah berikut. Masukkan nama klaster Anda dengan opsi `--name`, dan tentukan nama pasangan kunci EC2 Anda dengan opsi `--ec2-attributes`.

```
aws emr create-cluster \
--name "<My First EMR Cluster>" \
--release-label <emr-5.36.1> \
--applications Name=Spark \
--ec2-attributes KeyName=<myEMRKeyPairName> \
--instance-type m5.xlarge \
--instance-count 3 \
--use-default-roles
```

Perhatikan nilai lain yang diperlukan untuk `--instance-type`, `--instance-count`, dan `--use-default-roles`. Nilai-nilai ini telah dipilih untuk klaster tujuan umum. Untuk informasi selengkapnya`create-cluster`, lihat Referensi AWS CLI Perintah.

> **Note**
>
> Karakter lanjutan baris Linux (\) disertakan agar mudah dibaca Karakter ini bisa dihapus atau digunakan dalam perintah Linux. Untuk Windows, hapus atau ganti dengan tanda sisipan (^).

Anda akan melihat output seperti berikut. Output menunjukkan `ClusterId` dan `ClusterArn` klaster baru Anda. Perhatikan Anda`ClusterId`. Anda menggunakan `ClusterId` untuk memeriksa status cluster dan mengirimkan pekerjaan.

```
{
    "ClusterId": "myClusterId",
    "ClusterArn": "myClusterArn"
}
```

3. Periksa status klaster Anda dengan perintah berikut.

```
aws emr describe-cluster --cluster-id <myClusterId>
```

Anda akan melihat output seperti berikut dengan `Status` objek untuk cluster baru Anda.

```
{
    "Cluster": {
        "Id": "myClusterId",
        "Name": "My First EMR Cluster",
        "Status": {
            "State": "STARTING",
            "StateChangeReason": {
                "Message": "Configuring cluster software"
            }
        }
     }
}
```

`State`Nilai berubah dari `STARTING` `RUNNING` ke `WAITING` sebagai Amazon EMR menyediakan klaster.

Status klaster berubah menjadi **`WAITING`** saat klaster aktif, berjalan, dan siap menerima pekerjaan. Untuk informasi tentang status klaster, lihat Memahami siklus hidup klaster.

Langkah 2: Kelola klaster Amazon EMR

Mengirim pekerjaan ke Amazon EMR

Setelah Anda meluncurkan klaster, Anda dapat mengirimkan pekerjaan ke cluster yang sedang berjalan untuk memproses dan menganalisis data. Anda mengirimkan pekerjaan ke klaster Amazon EMR sebagai langkah. Langkah adalah unit kerja yang terdiri dari satu atau lebih tindakan. Misalnya, Anda dapat mengirimkan satu langkah untuk mengomputasi nilai, atau untuk mentransfer dan memproses data. Anda dapat mengirimkan langkah saat membuat klaster, atau ke klaster yang sedang berjalan. Di bagian tutorial ini, Anda mengirimkan `health_violations.py` sebagai langkah ke cluster yang sedang berjalan. Untuk mempelajari lebih lanjut tentang langkah-langkah, lihatKirim pekerjaan ke sebuah klaster.

New console

Untuk mengirimkan aplikasi Spark sebagai langkah dengan konsol baru

1. Masuk keAWS Management Console, dan buka konsol Amazon EMR di https://console.aws.amazon.com/emr.
2. Di bawah EMR pada EC2 di panel navigasi kiri, pilih Cluster, dan kemudian pilih cluster tempat Anda ingin mengirimkan pekerjaan. Keadaan cluster harus Menunggu.
3. Pilih tab Langkah, lalu pilih Tambahkan langkah.
4. Konfigurasikan langkah sesuai dengan pedoman berikut:
   - Untuk Tipe, pilih aplikasi Spark. Anda akan melihat bidang tambahan untuk mode Deploy, Lokasi aplikasi, dan opsi SPARK-submit.
   - Untuk Nama, masukkan nama baru. Jika Anda memiliki banyak langkah dalam sebuah klaster, penamaan setiap langkah membantu Anda melacak mereka.

- Untuk mode Deploy, biarkan mode Cluster nilai default. Untuk informasi selengkapnya tentang mode penyebaran Spark, lihat Ringkasan mode klaster dalam dokumentasi Apache Spark.
- Untuk lokasi Aplikasi, masukkan lokasi `health_violations.py` skrip Anda di Amazon S3, seperti *s3://DOC-EXAMPLE-BUCKET/health_violations.py.*
- Biarkan bidang opsi SPARK-submit kosong. Untuk informasi selengkapnya tentang `spark-submit` opsi, lihat Meluncurkan aplikasi dengan spark-submit.
- Dalam bidang Argumen, masukkan argumen dan nilai berikut:

```
--data_source s3://DOC-EXAMPLE-BUCKET/food_establishment_data.csv
--output_uri s3://DOC-EXAMPLE-BUCKET/myOutputFolder
```

  Ganti *s3://DOC-EXAMPLE-BUCKET/food_establishment_data.csv* dengan URI bucket S3 dari data input yang Anda siapkan. Menyiapkan aplikasi dengan data input untuk Amazon EMR

  Ganti *DOC-EXAMPLE-BUCKET* dengan nama bucket yang Anda buat untuk tutorial ini, dan ganti *myOutputFolder*dengan nama untuk folder output cluster Anda.

- Untuk Tindakan jika langkah gagal, terima opsi default Lanjutkan. Dengan cara ini, jika langkah gagal, cluster terus berjalan.

5. Pilih Tambahkan untuk mengirimkan langkah. Langkah akan ditampilkan di konsol dengan status Tertunda.
6. Pantau status langkah. Ini harus berubah dari Pending ke Running to Completed. Untuk menyegarkan status di konsol, pilih ikon refresh di sebelah kanan Filter. Skrip membutuhkan waktu sekitar satu menit untuk dijalankan. Ketika status berubah menjadi Selesai, langkah telah berhasil diselesaikan.

Old console

Untuk mengirimkan aplikasi Spark sebagai langkah dengan konsol lama

1. Arahkan ke konsol Amazon EMR baru dan pilih Beralih ke konsol lama dari navigasi samping. Untuk informasi selengkapnya tentang apa yang diharapkan saat Anda beralih ke konsol lama, lihat Menggunakan konsol lama.
2. Pilih nama klaster Anda dari Daftar Cluster. Keadaan cluster harus Menunggu.

3. Pilih Langkah, lalu pilih Tambahkan langkah.
4. Konfigurasikan langkah sesuai dengan pedoman berikut:

   - Untuk Jenis langkah, pilih Aplikasi Spark. Anda harus melihat bidang tambahan untuk Mode Deploy, Opsi Spark-submit, dan Lokasi aplikasi muncul.
   - Untuk Nama, biarkan nilai default atau ketik nama baru. Jika Anda memiliki banyak langkah dalam sebuah klaster, penamaan setiap langkah membantu Anda melacak mereka.
   - Untuk Mode deploy, biarkan nilai default Klaster. Untuk informasi selengkapnya tentang mode deployment Spark, lihat Gambaran umum mode klaster dalam dokumentasi Apache Spark.
   - Biarkan bidang Opsi Spark-submit tetap kosong. Untuk informasi selengkapnya tentang opsi `spark-submit`, lihat Meluncurkan aplikasi dengan spark-submit.
   - Untuk Lokasi aplikasi, masukkan lokasi skrip `health_violations.py` Anda di Amazon S3. Misalnya, *`s3://DOC-EXAMPLE-BUCKET/health_violations.py`*.
   - Dalam bidang Argumen, masukkan argumen dan nilai berikut:

     ```
     --data_source s3://DOC-EXAMPLE-BUCKET/food_establishment_data.csv
     --output_uri s3://DOC-EXAMPLE-BUCKET/myOutputFolder
     ```

     Ganti *`s3://DOC-EXAMPLE-BUCKET/food_establishment_data.csv`* dengan URI S3 dari data input yang Anda siapkan di Menyiapkan aplikasi dengan data input untuk Amazon EMR.

     Ganti *`DOC-EXAMPLE-BUCKET`* dengan nama bucket yang Anda buat untuk tutorial ini, dan *`myOutputFolder`*dengan nama untuk folder output cluster Anda.
   - Untuk Tindakan pada kegagalan, terima opsi default Lanjutkan sehingga jika langkah gagal, klaster bisa terus berjalan.
5. Pilih Tambahkan untuk mengirimkan langkah. Langkah akan ditampilkan di konsol dengan status Tertunda.
6. Periksa status langkah yang akan diubah dari Pending menjadi Running to Completed. Untuk menyegarkan status di konsol, pilih ikon refresh di sebelah kanan Filter. Skrip membutuhkan waktu sekitar satu menit untuk dijalankan.

Anda akan tahu langkah berhasil selesai ketika status berubah ke Selesai.

CLI

Untuk mengirimkan aplikasi Spark sebagai langkah dengan AWS CLI

1. Pastikan Anda memiliki `ClusterId` dari klaster yang Anda luncurkan di Meluncurkan klaster Amazon EMR. Anda juga dapat mengambil ID klaster dengan perintah berikut.

```
aws emr list-clusters --cluster-states WAITING
```

2. Kirim `health_violations.py` sebagai langkah dengan `add-steps` perintah dan Anda`ClusterId`.

   - Anda dapat menentukan nama untuk langkah Anda dengan mengganti *"Aplikasi Spark Saya"*. Di array `Args`, ganti *s3://DOC-EXAMPLE-BUCKET/health_violations.py* dengan lokasi aplikasi `health_violations.py` Anda.
   - Ganti *s3://DOC-EXAMPLE-BUCKET/food_establishment_data.csv* dengan lokasi S3 dari set data `food_establishment_data.csv` Anda.
   - Ganti *s3://DOC-EXAMPLE-BUCKET/ MyOutputFolder* dengan jalur S3 dari bucket yang Anda tunjuk dan nama untuk folder keluaran cluster Anda.
   - `ActionOnFailure=CONTINUE`berarti cluster terus berjalan jika langkah gagal.

```
aws emr add-steps \
--cluster-id <myClusterId> \
--steps Type=Spark,Name="<My Spark
 Application>",ActionOnFailure=CONTINUE,Args=[<s3://DOC-EXAMPLE-
BUCKET/health_violations.py>,--data_source,<s3://DOC-EXAMPLE-BUCKET/
food_establishment_data.csv>,--output_uri,<s3://DOC-EXAMPLE-BUCKET/
MyOutputFolder>]
```

Untuk informasi selengkapnya tentang mengirimkan langkah-langkah menggunakan CLI, lihat Referensi Perintah AWS CLI.

Setelah Anda mengirimkan langkah, Anda akan melihat output seperti berikut dengan daftar`StepIds`. Karena Anda mengirimkan satu langkah, Anda hanya akan melihat satu ID dalam daftar. Salin ID langkah Anda. Anda menggunakan ID langkah Anda untuk memeriksa status langkah.

```
{
```

```
    "StepIds": [
        "s-1XXXXXXXXXXA"
    ]
}
```

3. Query status langkah Anda dengan `describe-step` perintah.

```
aws emr describe-step --cluster-id <myClusterId> --step-id <s-1XXXXXXXXXXA>
```

Anda akan melihat output seperti berikut dengan informasi tentang langkah Anda.

```
{
    "Step": {
        "Id": "s-1XXXXXXXXXXA",
        "Name": "My Spark Application",
        "Config": {
            "Jar": "command-runner.jar",
            "Properties": {},
            "Args": [
                "spark-submit",
                "s3://DOC-EXAMPLE-BUCKET/health_violations.py",
                "--data_source",
                "s3://DOC-EXAMPLE-BUCKET/food_establishment_data.csv",
                "--output_uri",
                "s3://DOC-EXAMPLE-BUCKET/myOutputFolder"
            ]
        },
        "ActionOnFailure": "CONTINUE",
        "Status": {
            "State": "COMPLETED"
        }
    }
}
```

`State` dari langkah berubah dari `PENDING` ke `RUNNING` ke `COMPLETED` selagi langkah berjalan. Langkahnya membutuhkan waktu sekitar satu menit untuk dijalankan, jadi Anda mungkin perlu memeriksa statusnya beberapa kali.

Anda akan tahu langkah berhasil selesai ketika `State` berubah ke **`COMPLETED`**.

Untuk informasi lebih lanjut tentang siklus hidup langkah, lihat Menjalankan langkah-langkah untuk memproses data.

Melihat hasil

Setelah satu langkah berjalan dengan sukses, Anda dapat melihat hasil outputnya di folder keluaran Amazon S3 Anda.

Untuk melihat hasil **`health_violations.py`**

1. Buka konsol Amazon S3 di https://console.aws.amazon.com/s3/.
2. Pilih Nama bucket kemudian folder output yang Anda tentukan ketika Anda mengirimkan langkah. Misalnya, *`DOC-EXAMPLE-BUCKET`* dan kemudian. *`myOutputFolder`*
3. Verifikasi bahwa item berikut muncul di folder keluaran Anda:
   - Sebuah objek berukuran kecil yang disebut. `_SUCCESS`
   - File CSV yang dimulai dengan awalan `part-` yang berisi hasil Anda.
4. Pilih objek dengan hasil Anda, lalu pilih Unduh untuk menyimpan hasilnya ke sistem file lokal Anda.
5. Buka hasilnya di editor pilihan Anda. File keluaran mencantumkan sepuluh perusahaan makanan teratas dengan pelanggaran paling merah. File output juga menunjukkan jumlah total pelanggaran merah untuk setiap pendirian.

   Berikut ini adalah contoh `health_violations.py` hasil.

```
name, total_red_violations
SUBWAY, 322
T-MOBILE PARK, 315
WHOLE FOODS MARKET, 299
PCC COMMUNITY MARKETS, 251
TACO TIME, 240
MCDONALD'S, 177
THAI GINGER, 153
SAFEWAY INC #1508, 143
TAQUERIA EL RINCONSITO, 134
HIMITSU TERIYAKI, 128
```

Untuk informasi lebih lanjut tentang output klaster Amazon EMR, lihat Mengkonfigurasi lokasi output.

(Opsional) Hubungkan ke klaster Amazon EMR yang sedang berjalan

Saat Anda menggunakan Amazon EMR, Anda mungkin ingin terhubung ke klaster yang sedang berjalan untuk membaca file log, men-debug klaster, atau menggunakan alat CLI seperti shell Spark. Amazon EMR memungkinkan Anda terhubung ke klaster menggunakan protokol Secure Shell (SSH). Bagian ini mencakup cara mengkonfigurasi SSH, terhubung ke klaster Anda, dan melihat file log untuk Spark. Untuk informasi selengkapnya tentang menghubungkan ke klaster, lihatAutentikasi ke simpul klaster Amazon EMR.

Otorisasi koneksi SSH ke klaster Anda

Sebelum terhubung ke klaster, Anda perlu memodifikasi grup keamanan klaster untuk mengotorisasi koneksi SSH masuk. Grup keamanan Amazon EC2 bertindak sebagai firewall virtual untuk mengontrol lalu lintas masuk dan keluar ke klaster Anda. Saat Anda membuat klaster untuk tutorial ini, Amazon EMR membuat grup keamanan berikut atas nama Anda:

ElasticMapReduce-menguasai

Grup keamanan terkelola Amazon EMR default yang terkait dengan node utama. Dalam klaster Amazon EMR, node utama adalah instans Amazon EC2 yang mengelola klaster.

ElasticMapReduce-budak

Grup keamanan keamanan default yang terkait dengan simpul tugas dan core.

New console

Untuk mengizinkan akses SSH untuk sumber tepercaya untuk grup keamanan utama dengan konsol baru

Untuk mengedit grup keamanan, Anda harus memiliki izin untuk mengelola grup keamanan untuk VPC tempat klaster berada. Untuk informasi selengkapnya, lihat Mengubah Izin untuk pengguna dan Kebijakan Contoh yang memungkinkan pengelolaan grup keamanan EC2 di Panduan Pengguna IAM.

1. Masuk keAWS Management Console, dan buka konsol Amazon EMR di https://console.aws.amazon.com/emr.
2. Di bawah EMR pada EC2 di panel navigasi kiri, pilih Cluster, dan kemudian pilih cluster yang ingin Anda perbarui. Ini membuka halaman rincian klaster. Tab Properties pada halaman ini harus dipilih sebelumnya.

3. Di bawah Networking di tab Properties, pilih tanda panah di sebelah grup keamanan EC2 (firewall) untuk memperluas bagian ini. Di bawah Node utama, pilih tautan grup keamanan. Ketika Anda telah menyelesaikan langkah-langkah berikut, Anda dapat secara opsional kembali ke langkah ini, memilih Core dan task node, dan ulangi langkah-langkah berikut untuk memungkinkan akses klien SSH ke node inti dan tugas.
4. Ini membuka konsol EC2. Pilih tab Aturan masuk dan kemudian Edit aturan masuk.
5. Memeriksa aturan masuk yang mengizinkan akses publik dengan pengaturan berikut. Jika ada, pilih Hapus untuk menghapusnya.

   - Jenis

     SSH
   - Pelabuhan

     22
   - Sumber

     Kustom 0.0.0.0/0

   > ⚠ Warning
   >
   > Sebelum Desember 2020, grup keamanan ElasticMapReduce -master memiliki aturan yang telah dikonfigurasi sebelumnya untuk memungkinkan lalu lintas masuk di Port 22 dari semua sumber. Aturan ini dibuat untuk menyederhanakan koneksi SSH awal ke simpul utama. Kami sangat menyarankan agar Anda menghapus aturan masuk ini dan membatasi lalu lintas ke sumber tepercaya.

6. Gulir ke bagian bawah daftar aturan dan pilih Tambahkan Aturan.
7. Untuk Jenis, pilih SSH. Memilih SSH secara otomatis memasuki TCP untuk Protokol dan 22 untuk Port Range.
8. Untuk sumber, pilih IP saya untuk secara otomatis menambahkan alamat IP Anda sebagai alamat sumber. Anda juga dapat menambahkan berbagai alamat IP klien tepercaya khusus, atau membuat aturan tambahan untuk klien lain. Banyak lingkungan jaringan secara dinamis mengalokasikan alamat IP, jadi Anda mungkin perlu memperbarui alamat IP Anda untuk klien tepercaya di masa mendatang.
9. Pilih Save (Simpan).

10. Secara opsional, pilih Core dan task node dari daftar dan ulangi langkah-langkah di atas untuk memungkinkan akses klien SSH ke node inti dan tugas.

Old console

Untuk memberikan akses SSH sumber tepercaya ke grup keamanan utama dengan konsol lama

Untuk mengedit grup keamanan, Anda harus memiliki izin untuk mengelola grup keamanan untuk VPC tempat klaster berada. Untuk informasi selengkapnya, lihat Mengubah Izin untuk pengguna dan Kebijakan Contoh yang memungkinkan pengelolaan grup keamanan EC2 di Panduan Pengguna IAM.

1. Arahkan ke konsol Amazon EMR baru dan pilih Beralih ke konsol lama dari navigasi samping. Untuk informasi selengkapnya tentang apa yang diharapkan saat Anda beralih ke konsol lama, lihat Menggunakan konsol lama.
2. Pilih Klaster. Pilih Nama klaster yang ingin Anda modifikasi.
3. Pilih tautan Grup keamanan untuk Master di bawah Keamanan dan akses.
4. Pilih ElasticMapReduce-master dari daftar.
5. Pilih tab Aturan masuk dan kemudian Edit aturan masuk.
6. Memeriksa aturan masuk yang mengizinkan akses publik dengan pengaturan berikut. Jika ada, pilih Hapus untuk menghapusnya.

   - Jenis

     SSH
   - Pelabuhan

     22
   - Sumber

     Kustom 0.0.0.0/0

   > ⚠ Warning
   >
   > Sebelum Desember 2020, grup keamanan ElasticMapReduce -master memiliki aturan yang telah dikonfigurasi sebelumnya untuk memungkinkan lalu lintas masuk di Port 22 dari semua sumber. Aturan ini dibuat untuk menyederhanakan koneksi

SSH awal ke node utama. Kami sangat menyarankan agar Anda menghapus aturan masuk ini dan membatasi lalu lintas ke sumber tepercaya.

7. Gulir ke bagian bawah daftar aturan dan pilih Tambahkan Aturan.
8. Untuk Jenis, pilih SSH.

   Memilih SSH secara otomatis memasuki TCP untuk Protokol dan 22 untuk Port Range.
9. Untuk sumber, pilih IP saya untuk secara otomatis menambahkan alamat IP Anda sebagai alamat sumber. Anda juga dapat menambahkan berbagai alamat IP klien tepercaya khusus, atau membuat aturan tambahan untuk klien lain. Banyak lingkungan jaringan secara dinamis mengalokasikan alamat IP, jadi Anda mungkin perlu memperbarui alamat IP Anda untuk klien tepercaya di masa mendatang.
10. Pilih Save (Simpan).
11. Secara opsional, pilih ElasticMapReduce-slave dari daftar dan ulangi langkah-langkah di atas untuk memungkinkan akses klien SSH ke node inti dan tugas.

Hubungkan ke klaster Anda menggunakan AWS CLI

Terlepas dari sistem operasi Anda, Anda dapat membuat koneksi SSH ke klaster Anda menggunakan. AWS CLI

Untuk menyambung ke klaster Anda dan melihat file log menggunakan AWS CLI

1. Gunakan perintah berikut untuk membuka koneksi SSH ke klaster Anda. Ganti <mykeypair.key>dengan path lengkap dan nama file file pasangan kunci Anda. Sebagai contoh, `C:\Users\<username>\.ssh\mykeypair.pem`.

   ```
   aws emr ssh --cluster-id <j-2AL4XXXXXX5T9> --key-pair-file <~/mykeypair.key>
   ```
2. Arahkan `/mnt/var/log/spark` ke untuk mengakses log Spark pada master node klaster Anda. Kemudian lihat file di lokasi itu. Untuk daftar file log tambahan pada node master, lihatLihat file log pada simpul utama.

   ```
   cd /mnt/var/log/spark
   ls
   ```

Langkah 3: Bersihkan sumber daya Amazon EMR Anda

Mengakhiri klaster Anda

Sekarang setelah Anda mengirimkan pekerjaan ke klaster Anda dan melihat hasil PySpark aplikasi Anda, Anda dapat mengakhiri klaster. Mengakhiri klaster menghentikan semua biaya Amazon EMR terkait klaster dan instans Amazon EC2.

Saat Anda mengakhiri klaster, Amazon EMR mempertahankan metadata tentang klaster selama dua bulan tanpa biaya. Metadata yang diarsipkan membantu Anda mengkloning klaster untuk pekerjaan baru atau mengunjungi kembali konfigurasi klaster untuk tujuan referensi. Metadata tidak menyertakan data yang ditulis cluster ke S3, atau data yang disimpan dalam HDFS pada cluster.

> **Note**
>
> Konsol Amazon EMR tidak mengizinkan Anda menghapus klaster dari tampilan daftar setelah Anda mengakhiri klaster. Klaster yang diakhiri akan menghilang dari konsol ketika Amazon EMR membersihkan metadata.

New console

Untuk mengakhiri klaster dengan konsol baru

1. Masuk keAWS Management Console, dan buka konsol Amazon EMR di https://console.aws.amazon.com/emr.
2. Pilih Cluster, lalu pilih klaster yang ingin Anda hentikan.
3. Di bawah menu tarik-turun Tindakan, pilih Hentikan klaster.
4. Pilih Hentikan di kotak dialog. Tergantung pada konfigurasi cluster, penghentian dapat memakan waktu 5 hingga 10 menit. Untuk informasi selengkapnya tentang cara klaster Amazon EMR, lihat. Mengakhiri suatu klaster

Old console

Untuk mengakhiri cluster dengan konsol lama

1. Arahkan ke konsol Amazon EMR baru dan pilih Beralih ke konsol lama dari navigasi samping. Untuk informasi selengkapnya tentang apa yang diharapkan saat Anda beralih ke konsol lama, lihat Menggunakan konsol lama.
2. Pilih Cluster, lalu pilih klaster yang ingin Anda hentikan. Misalnya, *Klaster EMR Pertama Saya*.
3. Pilih Akhiri untuk membuka prompt Akhiri klaster.
4. Pilih Hentikan di prompt terbuka. Tergantung pada konfigurasi cluster, penghentian dapat memakan waktu 5 hingga 10 menit. Untuk informasi selengkapnya tentang mengakhiri klaster Amazon EMR, lihat. Mengakhiri suatu klaster

> **Note**
>
> Jika Anda mengikuti tutorial dengan saksama, perlindungan penghentian harus dimatikan. Perlindungan terminasi klaster mencegah penghentian yang tidak disengaja. Jika perlindungan penghentian aktif, Anda akan melihat prompt untuk mengubah pengaturan sebelum mengakhiri klaster. Pilih Ubah, kemudian Nonaktif.

CLI

Untuk mengakhiri cluster dengan AWS CLI

1. Memulai proses terminasi cluster dengan perintah berikut. Ganti *< myClusterId >* dengan ID cluster sampel Anda. Perintah tidak mengembalikan output.

```
aws emr terminate-clusters --cluster-ids <myClusterId>
```

2. Untuk memeriksa apakah proses terminasi klaster sedang berlangsung, periksa status klaster dengan perintah berikut.

```
aws emr describe-cluster --cluster-id <myClusterId>
```

Berikut ini adalah contoh output dalam format JSON. Klaster `Status` akan berubah dari **`TERMINATING`** ke **`TERMINATED`**. Pengakhiran dapat memakan waktu 5 hingga 10 menit

tergantung pada konfigurasi klaster Anda. Untuk informasi selengkapnya tentang mengakhiri klaster Amazon EMR, lihat. Mengakhiri suatu klaster

```
{
    "Cluster": {
        "Id": "j-xxxxxxxxxxxx",
        "Name": "My Cluster Name",
        "Status": {
            "State": "TERMINATED",
            "StateChangeReason": {
                "Code": "USER_REQUEST",
                "Message": "Terminated by user request"
            }
        }
    }
}
```

Menghapus sumber daya S3

Untuk menghindari biaya tambahan, Anda harus menghapus bucket Amazon S3 Anda. Menghapus bucket akan menghapus semua sumber daya Amazon S3 untuk tutorial ini. Bucket Anda harus berisi:

- PySparkSkrip
- Dataset masukan
- Folder hasil keluaran Anda
- Folder file log Anda

Anda mungkin perlu mengambil langkah tambahan untuk menghapus file yang disimpan jika Anda menyimpan PySpark skrip atau output Anda di lokasi yang berbeda.

> **Note**
>
> Klaster Anda harus dihentikan sebelum Anda menghapus bucket Anda. Jika tidak, Anda mungkin tidak diizinkan untuk mengosongkan ember.

Untuk menghapus bucket Anda, ikuti petunjuk di Bagaimana cara menghapus bucket S3? dalam Panduan Pengguna Amazon Simple Storage Service.

Langkah selanjutnya

Anda sekarang telah meluncurkan klaster Amazon EMR pertama Anda dari awal hingga akhir. Anda juga telah menyelesaikan tugas-tugas penting ESDM seperti mempersiapkan dan mengirimkan aplikasi big data, melihat hasil, dan mengakhiri klaster.

Gunakan topik berikut untuk mempelajari lebih lanjut tentang bagaimana Anda dapat menyesuaikan alur kerja Amazon EMR Anda.

Menjelajahi aplikasi big data untuk Amazon EMR

Temukan dan bandingkan aplikasi big data yang dapat Anda instal pada klaster dalam Panduan Rilis Amazon EMR. Panduan Rilis merinci setiap versi rilis EMR dan menyertakan tips untuk menggunakan kerangka kerja seperti Spark dan Hadoop di Amazon EMR.

Merencanakan perangkat keras, jaringan, dan keamanan klaster

Dalam tutorial ini, Anda membuat cluster EMR sederhana tanpa mengkonfigurasi opsi lanjutan. Opsi lanjutan memungkinkan Anda menentukan jenis instans Amazon EC2, jaringan klaster, dan keamanan klaster. Untuk informasi lebih lanjut tentang perencanaan dan peluncuran klaster yang memenuhi persyaratan Anda, lihat Merencanakan dan mengonfigurasi klaster dan. Keamanan di Amazon EMR

Mengelola klaster

Menyelam lebih dalam ke bekerja dengan menjalankan cluster diMengelola klaster. Untuk mengelola klaster, Anda dapat terhubung ke klaster, langkah debug, dan melacak aktivitas dan kesehatan klaster. Anda juga dapat menyesuaikan sumber daya klaster sebagai respons terhadap tuntutan beban kerja dengan penskalaan terkelola ESDM.

Menggunakan antarmuka yang berbeda

Selain konsol Amazon EMR, Anda dapat mengelola Amazon EMR menggunakan AWS Command Line Interface, API layanan web, atau salah satu dari AWS SDK yang banyak didukung. Untuk informasi selengkapnya, lihat Antarmuka manajemen.

Anda juga dapat berinteraksi dengan aplikasi yang diinstal pada klaster Amazon EMR dengan berbagai cara. Beberapa aplikasi seperti Apache Hadoop mempublikasikan antarmuka web yang

dapat Anda lihat. Untuk informasi selengkapnya, lihat Melihat antarmuka web yang di-host pada klaster Amazon EMR.

Menelusuri blog teknis EMR

Untuk panduan sampel dan diskusi teknis mendalam tentang fitur Amazon EMR baru, lihat blog big data. AWS

Apa yang baru dengan konsol?

Amazon EMR telah bermigrasi ke pengalaman baru. Konsol baru ini menawarkan antarmuka yang diperbarui yang memberi Anda cara intuitif untuk mengelola lingkungan EMR Amazon Anda dan memberi Anda akses mudah ke dokumentasi, informasi produk, dan sumber daya lainnya. Halaman ini menjelaskan perbedaan penting antara pengalaman konsol lama dan yang baru AWS Management Console untuk Amazon EMR.

Konsol apa saya?

Untuk menentukan konsol EMR Amazon yang saat ini Anda gunakan, lihat URL untuk halaman konsol di browser Anda:

- URL konsol baru - https://console.aws.amazon.com/emr
- URL konsol lama - https://console.aws.amazon.com/elasticmapreduce

> **Note**
>
> Amazon EMR memiliki pengalaman konsol baru. Konsol lama telah usang dan tidak lagi tersedia.

Fungsionalitas konsol Amazon EMR bermigrasi ke pengalaman baru secara bertahap. Tabel berikut mencantumkan komponen konsol EMR Amazon utama dan status migrasi konsolnya.

Komponen konsol Amazon EMR	Konsol baru	Konsol lama
Studio EMR 1	✓	✓
Buat dan kelola cluster	✓	✓
Blokir akses publik	✓	✓
Pantau CloudWatch Acara Amazon	✓	✓

Komponen konsol Amazon EMR	Konsol baru	Konsol lama
Konfigurasi grup keamanan	✓	✓
Cluster virtual (Amazon EMR di EKS)	✓	✓
Lihat dan kelola subnet Amazon Virtual Private Cloud 2	✓	✓
Notebook 3	✓	✓

[1] EMR Studio menggunakan pengalaman antarmuka baru di konsol baru dan lama.

[2] Di konsol baru, Anda dapat melihat dan mengelola subnet VPC Amazon Anda di dalam bagian Jaringan saat Anda membuat cluster. Di konsol lama, gunakan tautan di bilah navigasi sebelah kiri untuk mengakses daftar subnet Amazon VPC.

[3] EMR Notebooks tersedia sebagai EMR Studio Workspaces di konsol baru. Anda masih dapat menggunakan notebook yang ada di konsol lama, tetapi Anda tidak dapat membuat notebook baru di konsol lama. Tombol Create Workspace di konsol baru menggantikan fungsi ini. Untuk mengakses atau membuat Ruang Kerja, pengguna EMR Notebooks memerlukan izin peran IAM tambahan. Untuk informasi selengkapnya, lihat Amazon EMR Notebook adalah Amazon EMR Studio Workspaces di konsol baru dan Apa yang baru di konsol?

Menggunakan konsol lama

Amazon EMR memiliki pengalaman konsol baru. Konsol lama sudah usang dan tidak lagi tersedia.

Ringkasan perbedaan

Bagian ini menguraikan perbedaan antara konsol EMR Amazon lama dan pengalaman konsol EMR Amazon yang baru. Perbedaannya termasuk dalam kategori berikut:

- Kompatibilitas cluster antara konsol lama dan baru
- Perbedaan saat Anda membuat cluster

- Perbedaan saat Anda melihat atau mengedit detail klaster
- Perbedaan saat Anda mendaftar dan mencari cluster
- Perbedaan saat Anda bekerja dengan konfigurasi keamanan

Kompatibilitas cluster antara konsol lama dan baru

Dalam beberapa kasus, cluster yang Anda buat di konsol EMR Amazon lama mungkin tidak kompatibel dengan konsol baru. Daftar berikut menjelaskan persyaratan kompatibilitas untuk konsol EMR Amazon baru.

- Konsol baru ini mendukung cluster yang dibuat di Amazon EMR rilis 5.20.1 dan yang lebih baru.
- Anda dapat mengkloning cluster yang menggunakan penskalaan otomatis di konsol baru, tetapi Anda hanya dapat membuat klaster baru jika Anda ingin menskalakannya secara manual atau menggunakan penskalaan terkelola.

Untuk membuat dan bekerja dengan cluster yang tidak kompatibel dengan konsol baru, Anda dapat menggunakan AWS Command Line Interface (AWS CLI), AWS SDK, atau konsol lama.

Perbedaan saat Anda membuat cluster

Tabel berikut menyoroti perbedaan yang dapat Anda harapkan saat membuat cluster dengan konsol EMR Amazon baru sebagai lawan dari konsol EMR Amazon lama.

Kemampuan	Konsol baru	Konsol lama
Terminologi: Jenis simpul kluster EMR Amazon	Primer, inti, tugas	Guru, inti, tugas
Amazon EMR mendukung rilis 1	Amazon EMR rilis 5.20.1 dan yang lebih baru	Semua rilis Amazon EMR
Meluncurkan cluster dengan cepat	Gunakan tombol Create cluster di bawah panel Summary	Gunakan halaman Create cluster - Quick Options

Kemampuan	Konsol baru	Konsol lama
Mengonfigurasi batas waktu penyediaan Spot	Tentukan periode waktu tunggu untuk penyediaan instance untuk setiap armada di klaster Anda.	Anda tidak dapat menyesuaikan batas waktu penyediaan saat membuat klaster.
Peran layanan dan peran profil instans Amazon EC2	Konsol baru tidak membuat peran default; Anda harus membuat peran dengan Konsol IAM atau memilih peran IAM yang sudah dibuat	Mendukung pembuatan peran default dengan kebijakan v1 dan v2, atau Anda dapat memilih peran IAM yang sudah dibuat
Visibilitas cluster	Dari dalam konsol EMR Amazon, Anda tidak dapat membuat klaster terlihat oleh semua pengguna; kebijakan IAM Anda menentukan akses klaster	Dari dalam konsol EMR Amazon, Anda dapat membuat klaster terlihat oleh semua pengguna jika Anda menggunakan kebijakan pembuatan peran v1 yang tidak digunakan lagi
Jaringan - mengkonfigurasi subnet pribadi	Anda harus mengonfigurasi titik akhir Amazon S3 dan gateway NAT dari konsol Amazon S3 dan Amazon VPC masing-masing	Anda dapat mengonfigurasi titik akhir Amazon S3 dan gateway NAT langsung dari alur kerja Create cluster di konsol lama
Tampilan konsisten Sistem File EMR (EMRFS CV)	Dengan dirilisnya read-after-write konsistensi kuat Amazon S3 pada 1 Desember 2020, Anda tidak perlu menggunakan CV EMRFS dengan kluster EMR Anda	CV EMRFS diaktifkan, tetapi Anda dapat menonaktifkan CV EMRFS dan menghapus database Amazon DynamoDB yang digunakannya; lihat Tampilan konsisten untuk informasi selengkapnya

Kemampuan	Konsol baru	Konsol lama
Debugging	Anda dapat men-debug pekerjaan menggunak an antarmuka UI Aplikasi di halaman detail cluster	Anda dapat menggunakan alat debugger (langkah 3 dalam opsi lanjutan) untuk men-debug pekerjaan untuk cluster yang berjalan di Amazon EMR rilis 4.1.0 hingga 5.27.0

[1] Anda tidak dapat membuat atau mengedit cluster menggunakan rilis lebih awal dari Amazon EMR 5.20.1 di konsol baru, tetapi cluster yang ada yang dibuat menggunakan rilis lebih awal dari 5.20.1 akan terus berfungsi. Untuk membuat dan mengedit cluster dengan rilis Amazon EMR lebih awal dari 5.20.1, gunakan API atau CLI, atau beralih kembali ke konsol lama.

Perbedaan saat Anda mendaftar dan mencari cluster

Tabel berikut menyoroti perbedaan yang dapat Anda harapkan ketika Anda melihat dan mencari cluster dalam tampilan daftar dengan konsol EMR Amazon baru sebagai lawan dari konsol EMR Amazon lama.

> **Note**
>
> Untuk konsol lama dan baru, ketika Anda menerapkan filter data ke daftar cluster, itu menanyakan seluruh database. Tetapi ketika Anda memasukkan string teks ke dalam kotak pencarian, pencarian hanya berlaku untuk hasil yang daftar telah dimuat sisi klien.

Kemampuan	Konsol baru	Konsol lama
Melihat detail cluster	Anda dapat memilih ID Cluster untuk melihat detail klaster lengkap seperti opsi konfigurasi, UI aplikasi persisten, dan log.	Anda dapat memperluas dan menciutkan setiap baris cluster untuk melihat informasi seperti detail konfigurasi dan untuk mengakses tautan untuk pemantauan dan log klaster.

Kemampuan	Konsol baru	Konsol lama
Mencari cluster	Gunakan satu bidang pencarian untuk memasukkan kueri pencarian teks dan untuk membuat dan menerapka n filter data seperti “Status = Status aktif apa pun”.	Gunakan dropdown untuk menyaring status cluster (Aktif, Terminasi, Gagal) dan bidang terpisah untuk memasukka n kueri penelusuran teks.
Menemukan cluster yang gagal	Untuk mencari cluster yang gagal, terapkan filter Status = Diakhiri dengan kesalahan.	Untuk mencari klaster yang gagal, terapkan filter Cluster gagal.

Perbedaan saat Anda melihat atau mengedit detail klaster

Tabel berikut menyoroti perbedaan yang dapat Anda harapkan saat melihat atau mengedit detail untuk cluster yang ada dengan konsol EMR Amazon baru sebagai lawan dari konsol EMR Amazon lama.

Kemampuan	Konsol baru	Konsol lama
Melihat instans di grup instans dan armada instans Anda, bersama dengan opsi penskalaan, penyediaan, pengubahan ukuran, dan penghentian	Lihat opsi dan detail instance di tab Instances. Lihat opsi penghentian di tab Properti.	Lihat konfigurasi instans dan opsi penghentian di tab Perangkat Keras.
Melihat UI aplikasi, log, dan konfigurasi (Apache Spark UI, layanan Sejarah Spark, Apache Tez UI, server timeline YARN)	Lihat konfigurasi cluster di tab Konfigurasi. Luncurkan UI aplikasi live, persisten , untuk melihat log untuk aplikasi dari tab Applications.	Lihat konfigurasi cluster di tab Konfigurasi. Luncurkan UI aplikasi langsung, persisten, untuk melihat log untuk aplikasi dari tab antarmuka pengguna Aplikasi. Pada Januari 2023,

Kemampuan	Konsol baru	Konsol lama
		riwayat aplikasi tingkat tinggi tidak lagi tersedia.
Mengekspor cluster ke CLI	Opsi tersedia dari detail cluster dan tampilan daftar Menu tindakan sebagai “Lihat perintah untuk kloning kloning”	Opsi yang tersedia dari daftar cluster Lihat menu Tindakan sebagai "AWS CLIEkspor”

Perbedaan saat Anda bekerja dengan konfigurasi keamanan

Tabel berikut menyoroti perbedaan yang dapat Anda harapkan saat mengonfigurasi opsi keamanan dengan konsol EMR Amazon baru yang bertentangan dengan konsol EMR Amazon lama.

Kemampuan	Konsol baru	Konsol lama
Konfigurasi keamanan kloning	✓	
Tata kelola federasi menggunakan Trino dan Apache Ranger	✓	
Menggunakan peran runtime untuk mengirimkan pekerjaan ke klaster 1	✓	
Mengotorisasi akses ke data EMR File System (EMRFS)	Amazon S3 Access Points	AWS Identity and Access ManagementPeran (IAM)
AWS Lake Formationkontrol akses	Peran runtime	Federasi SAFL

[1] Untuk meneruskan peran selama pengiriman langkah, klaster Anda harus menggunakan konfigurasi keamanan dengan kebijakan izin IAM yang dilampirkan sehingga pengguna hanya dapat meneruskan peran yang disetujui dan pekerjaan Anda dapat mengakses sumber daya Amazon EMR. Untuk informasi selengkapnya, lihat Peran runtime untuk langkah-langkah EMR Amazon.

Amazon EMR Studio

Amazon EMR Studio adalah lingkungan pengembangan terintegrasi berbasis web (IDE) untuk notebook Jupyter yang dikelola sepenuhnya yang berjalan di kluster EMR Amazon. Anda dapat menyiapkan EMR Studio untuk tim Anda untuk mengembangkan, memvisualisasikan, dan men-debug aplikasi yang ditulis dalam R, Python, Scala, dan. PySpark EMR Studio terintegrasi dengan AWS Identity and Access Management (IAM) dan IAM Identity Center sehingga pengguna dapat masuk menggunakan kredensyal perusahaan mereka.

Anda dapat membuat EMR Studio tanpa biaya. Berlaku biaya untuk penyimpanan Amazon S3 dan untuk klaster Amazon EMR berlaku ketika Anda menggunakan EMR Studio. Untuk detail dan sorotan produk, lihat halaman layanan untuk Amazon EMR Studio.

Fitur utama dari EMR Studio

Amazon EMR Studio menyediakan fitur-fitur berikut:

- Mengautentikasi pengguna dengan AWS Identity and Access Management (IAM), atau AWS IAM Identity Center dengan atau tanpa propagasi identitas tepercaya dan penyedia identitas perusahaan Anda.
- Akses dan luncurkan klaster EMR Amazon sesuai permintaan untuk menjalankan pekerjaan Jupyter Notebook.
- Connect ke Amazon EMR di kluster EKS untuk mengirimkan pekerjaan saat pekerjaan berjalan.
- Jelajahi dan simpan contoh notebook. Untuk informasi selengkapnya tentang contoh buku catatan, lihat repositori contoh Notebook GitHub EMR Studio.
- Analisis data menggunakan Python,, Spark PySpark Scala, Spark R, atau SparkSQL, dan instal kernel dan pustaka khusus.
- Berkolaborasi secara real time dengan pengguna lain di Workspace yang sama. Untuk informasi selengkapnya, lihat Konfigurasikan kolaborasi Workspace.
- Gunakan EMR Studio SQL Explorer untuk menelusuri katalog data Anda, menjalankan kueri SQL, dan mengunduh hasil sebelum Anda bekerja dengan data di buku catatan.
- Jalankan notebook berparameter sebagai bagian dari alur kerja terjadwal dengan alat orkestrasi seperti Apache Airflow atau Amazon Managed Workflows for Apache Airflow. Untuk informasi selengkapnya, lihat Orchestrating analytics jobs on EMR Notebooks using MWAA dalam Blog Big Data AWS.

- Tautkan repositori kode seperti GitHub dan. BitBucket
- Melacak dan men-debug pekerjaan menggunakan Spark History Server, Tez UI, atau server timeline YARN.

EMR Studio juga memenuhi syarat HIPAA dan disertifikasi di bawah HITRUST CSF dan SOC 2. Untuk informasi selengkapnya tentang kepatuhan HIPAA untuk AWS layanan, lihathttps://aws.amazon.com/compliance/hipaa-compliance/. Untuk mempelajari lebih lanjut tentang kepatuhan CSF HITRUST untuk AWS layanan, lihat. https://aws.amazon.com/compliance/hitrust/ Untuk informasi selengkapnya tentang program kepatuhan lainnya untuk AWS layanan, lihat AWSLayanan dalam Lingkup menurut Program Kepatuhan.

Riwayat fitur Amazon EMR Studio

Tabel ini mencantumkan pembaruan untuk kemampuan penskalaan terkelola Amazon EMR.

Tanggal rilis	Kemampuan
November 26, 2023	Menambahkan dukungan untuk propagasi identitas tepercaya untuk EMR Studio dengan autentikasi IAM Identity Center.
Oktober 26, 2023	Ditambahkan kemampuan untuk membuat aplikasi EMR Serverless dengan kemampuan interaktif.
Februari 28, 2023	Menambahkan dukungan kunci AWS KMS yang dikelola pelanggan untuk penyimpanan log aplikasi untuk aplikasi EMR Tanpa Server.
Februari 23, 2023	Menambahkan pembuatan peran IAM satu klik untuk pengirima n pekerjaan EMR Tanpa Server. Menambahkan pencarian ECR saat Anda memilih gambar khusus untuk aplikasi EMR Tanpa Server.
Januari 27, 2023	Notebook eksekusi tanpa kepala dapat melacak kemajuan setiap eksekusi sel dengan `%execute_notebook` sihir.
Januari 23, 2023	Aplikasi persisten telah dioptimalkan untuk waktu peluncuran yang lebih cepat.

Cara Kerja Amazon EMR Studio

Amazon EMR Studio adalah sumber daya EMR Amazon yang Anda buat untuk tim pengguna. Setiap Studio adalah lingkungan pengembangan terintegrasi berbasis web mandiri untuk notebook Jupyter yang berjalan di kluster EMR Amazon. Pengguna masuk ke Studio menggunakan kredensi perusahaan.

Setiap EMR Studio yang Anda buat menggunakan sumber daya berikut: AWS

- Amazon Virtual Private Cloud (VPC) dengan subnet - Pengguna menjalankan kernel Studio dan aplikasi di Amazon EMR dan Amazon EMR pada kluster EKS di VPC yang ditentukan. EMR Studio dapat terhubung ke klaster apa pun di subnet yang Anda tentukan saat membuat Studio.
- Peran IAM dan kebijakan izin - Untuk mengelola izin pengguna, Anda membuat kebijakan izin IAM yang Anda lampirkan ke identitas IAM pengguna atau ke peran pengguna. EMR Studio juga menggunakan peran layanan IAM dan kelompok keamanan untuk berinteraksi dengan layanan lain. AWS Untuk informasi selengkapnya, silakan lihat Kontrol akses dan Menentukan grup keamanan untuk mengontrol lalu lintas jaringan EMR Studio.
- Grup keamanan - EMR Studio menggunakan grup keamanan untuk membuat saluran jaringan aman antara Studio dan cluster EMR.
- Lokasi cadangan Amazon S3 - EMR Studio menyimpan pekerjaan notebook di lokasi Amazon S3.

Langkah-langkah berikut menguraikan cara membuat dan mengelola EMR Studio:

1. Buat Studio di Anda Akun AWS dengan autentikasi IAM atau IAM Identity Center. Untuk petunjuk, silakan lihat Menyiapkan Amazon EMR Studio.
2. Tetapkan pengguna dan grup ke Studio Anda. Gunakan kebijakan izin untuk menetapkan izin berbutir halus untuk setiap pengguna. Untuk informasi lebih lanjut, lihat topiknyaMenetapkan dan mengelola pengguna EMR Studio.
3. Mulai memantau tindakan EMR Studio dengan AWS CloudTrail acara. Untuk informasi selengkapnya, lihat Memantau tindakan Amazon EMR Studio.
4. Berikan lebih banyak opsi klaster kepada pengguna Studio dengan templat klaster dan Amazon EMR di titik akhir yang dikelola EKS.

Otentikasi dan login pengguna

Amazon EMR Studio mendukung dua mode otentikasi: mode otentikasi IAM dan mode otentikasi IAM Identity Center. Mode IAM menggunakan AWS Identity and Access Management (IAM), sedangkan mode IAM Identity Center menggunakan. AWS IAM Identity Center Saat membuat EMR Studio, Anda memilih mode otentikasi untuk semua pengguna Studio tersebut.

Mode otentikasi IAM

Dengan mode otentikasi IAM, Anda dapat menggunakan otentikasi IAM atau federasi IAM.

Autentikasi IAM memungkinkan Anda mengelola identitas IAM seperti pengguna, grup, dan peran di IAM. Anda memberi pengguna akses ke Studio dengan kebijakan izin IAM dan kontrol akses berbasis atribut (ABAC).

Federasi IAM memungkinkan Anda membangun kepercayaan antara penyedia identitas pihak ketiga (iDP) AWS dan sehingga Anda dapat mengelola identitas pengguna melalui IDP Anda.

Mode otentikasi Pusat Identitas IAM

Mode autentikasi IAM Identity Center memungkinkan Anda memberi pengguna akses federasi ke EMR Studio. Anda dapat menggunakan IAM Identity Center untuk mengautentikasi pengguna dan grup dari direktori IAM Identity Center, direktori perusahaan yang ada, atau IDP eksternal seperti Azure Active Directory (AD). Anda kemudian mengelola pengguna dengan penyedia identitas Anda (iDP).

EMR Studio mendukung penggunaan penyedia identitas berikut untuk IAM Identity Center:

- AWS Managed Microsoft AD dan Direktori Aktif yang dikelola sendiri — Untuk informasi lebih lanjut, lihat Menghubungkan ke direktori Microsoft AD.
- Penyedia berbasis SAML – Untuk daftar lengkap, lihat Penyedia identitas yang didukung.
- Direktori Pusat Identitas IAM — Untuk informasi selengkapnya, lihat Mengelola identitas di Pusat Identitas IAM dan propagasi identitas tepercaya di seluruh aplikasi dalam Panduan Pengguna. AWS IAM Identity Center

Bagaimana otentikasi memengaruhi login dan penetapan pengguna

Mode autentikasi yang Anda pilih untuk EMR Studio memengaruhi cara pengguna masuk ke Studio, cara Anda menetapkan pengguna ke Studio, dan cara Anda mengotorisasi (memberikan izin kepada) pengguna untuk melakukan tindakan seperti membuat kluster EMR Amazon baru.

Tabel berikut merangkum metode login untuk EMR Studio sesuai dengan modus otentikasi.

Opsi login EMR Studio dengan mode otentikasi

Mode autentikasi	Metode login	Deskripsi
• IAM (otentikasi dan federasi) • Pusat Identitas IAM	URL Studio EMR	Pengguna masuk ke Studio menggunakan URL akses Studio. Sebagai contoh, `https://xxxxxxxxxxxxxxxxxxxxx.emrstudio-prod.us-east-1.amazonaws.com` . Pengguna memasukkan kredensi IAM saat Anda menggunakan autentikasi IAM. Saat Anda menggunakan federasi IAM atau IAM Identity Center, EMR Studio mengalihkan pengguna ke URL masuk penyedia identitas Anda untuk memasukkan kredensyal. Dalam konteks federasi identitas, opsi login ini disebut Service Provider (SP) memulai sign-in.
• IAM (federasi) • Pusat Identitas IAM	Portal penyedia identitas (iDP)	Pengguna masuk ke portal penyedia identitas Anda, seperti portal Azure, dan meluncurkan konsol EMR Amazon. Setelah meluncurkan konsol EMR Amazon, pengguna memilih dan membuka Studio dari daftar Studios. Anda juga dapat mengonfigurasi EMR Studio sebagai aplikasi SAMP sehingga pengguna dapat masuk ke Studio tertentu dari portal penyedia identitas Anda. Untuk petunjuk, lihat Untuk mengonfigurasi EMR Studio sebagai aplikasi SAMP di portal iDP Anda.

Mode autentikasi	Metode login	Deskripsi
		Dalam konteks federasi identitas, opsi login ini disebut penyedia identitas (iDP) memulai login.
• IAM (otentikasi)	AWS Management Console	Pengguna masuk ke AWS Management Console menggunakan kredensyal IAM dan membuka Studio dari daftar Studios di konsol EMR Amazon.

Tabel berikut menguraikan penugasan pengguna dan otorisasi untuk EMR Studio dengan mode otentikasi.

Penugasan dan otorisasi pengguna EMR Studio dengan mode otentikasi

Mode autentikasi	Penugasan pengguna	Otorisasi pengguna
IAM (otentikasi dan federasi)	Izinkan `CreateStudioPresignedUrl` tindakan dalam kebijakan izin IAM yang dilampirkan ke identitas IAM (pengguna, grup, atau peran). Untuk pengguna federasi, izinkan `CreateStudioPresignedUrl` tindakan dalam IAM dalam kebijakan izin yang Anda konfigurasikan untuk peran IAM yang Anda gunakan untuk federasi. Gunakan kontrol akses berbasis atribut (ABAC) untuk menentukan Studio atau Studio yang dapat diakses pengguna. Untuk petunjuk, silakan lihat Menetapkan pengguna atau grup ke EMR Studio.	Tentukan kebijakan izin IAM yang memungkinkan tindakan EMR Studio tertentu. Untuk pengguna asli, lampirkan kebijakan izin IAM ke identitas IAM (pengguna, grup, atau peran). Untuk pengguna federasi, izinkan tindakan Studio dalam kebijakan izin yang Anda konfigurasikan untuk peran IAM yang Anda gunakan untuk federasi. Untuk informasi selengkapnya, lihat Konfigurasikan izin pengguna EMR Studio untuk Amazon EC2 atau Amazon EKS.

Mode autentikasi	Penugasan pengguna	Otorisasi pengguna
Pusat Identitas IAM	Untuk Studios yang dibuat dengan `IdCUserAssignment` set to`REQUIRED`, petakan pengguna ke Studio dengan kebijakan sesi tertentu. Untuk informasi selengkapnya, lihat Menetapkan pengguna atau grup ke EMR Studio. Untuk Studio yang dibuat dengan `IdCUserAssignment` set to`OPTIONAL`, setiap pengguna atau grup Pusat Identitas dapat mengakses Studio.	Opsional: Tentukan kebijakan sesi IAM yang memungkinkan tindakan EMR Studio tertentu. Memetakan kebijakan sesi ke pengguna saat Anda menetapkan pengguna ke Studio. Untuk informasi selengkapnya, lihat Izin pengguna untuk mode otentikasi Pusat Identitas IAM.

Kontrol akses

Di Amazon EMR Studio, Anda mengonfigurasi otorisasi pengguna (izin) dengan kebijakan berbasis identitas AWS Identity and Access Management (IAM). Dalam kebijakan ini, Anda menentukan tindakan dan sumber daya yang diizinkan, serta kondisi di mana tindakan diizinkan.

Izin pengguna untuk mode otentikasi IAM

Untuk menetapkan izin pengguna saat Anda menggunakan autentikasi IAM untuk EMR Studio, Anda mengizinkan tindakan seperti `elasticmapreduce:RunJobFlow` dalam kebijakan izin IAM. Anda dapat membuat satu atau beberapa kebijakan izin untuk digunakan. Misalnya, Anda dapat membuat kebijakan dasar yang tidak mengizinkan pengguna membuat kluster EMR Amazon baru, dan kebijakan lain yang mengizinkan pembuatan klaster. Untuk daftar semua tindakan Studio, lihatAWS Identity and Access Managementizin untuk pengguna EMR Studio.

Izin pengguna untuk mode otentikasi Pusat Identitas IAM

Bila Anda menggunakan autentikasi IAM Identity Center, Anda membuat satu peran pengguna EMR Studio. Peran pengguna adalah peran IAM khusus yang diasumsikan Studio saat pengguna masuk.

Anda melampirkan kebijakan sesi IAM ke peran pengguna EMR Studio. Kebijakan sesi adalah jenis khusus dari kebijakan izin IAM yang membatasi apa yang dapat dilakukan pengguna federasi selama sesi login Studio. Kebijakan sesi memungkinkan Anda menetapkan izin khusus untuk pengguna atau grup tanpa membuat beberapa peran pengguna untuk EMR Studio.

Saat menetapkan pengguna dan grup ke Studio, Anda memetakan kebijakan sesi ke pengguna atau grup tersebut untuk menerapkan izin berbutir halus. Anda juga dapat memperbarui kebijakan sesi pengguna atau grup kapan saja. Amazon EMR menyimpan setiap pemetaan kebijakan sesi yang Anda buat.

Untuk informasi selengkapnya tentang kebijakan sesi, lihat Izin dan kebijakan dalam Panduan Pengguna AWS Identity and Access Management.

Workspace

Workspace adalah blok bangunan utama Amazon EMR Studio. Untuk mengatur buku catatan, pengguna membuat satu atau beberapa Ruang Kerja di Studio. Untuk informasi selengkapnya, lihat Pelajari dasar-dasar Ruang Kerja.

Mirip dengan ruang kerja di JupyterLab, Workspace mempertahankan status kerja notebook. Namun, antarmuka pengguna Workspace memperluas JupyterLabantarmuka sumber terbuka dengan alat tambahan untuk memungkinkan Anda membuat dan melampirkan kluster EMR, menjalankan pekerjaan, menjelajahi contoh notebook, dan menautkan repositori Git.

Daftar berikut mencakup fitur utama EMR Studio Workspaces:

- Visibilitas Workspace berbasis Studio. Ruang kerja yang Anda buat di satu Studio tidak terlihat di Studio lain.
- Secara default, Workspace dibagikan dan dapat dilihat oleh semua pengguna Studio. Namun, hanya satu pengguna yang dapat membuka dan bekerja di Workspace pada satu waktu. Untuk bekerja secara bersamaan dengan pengguna lain, Anda bisa Konfigurasikan kolaborasi Workspace
- Anda dapat berkolaborasi secara bersamaan dengan pengguna lain di Workspace saat Anda mengaktifkan kolaborasi Workspace. Untuk informasi selengkapnya, lihat Konfigurasikan kolaborasi Workspace.
- Notebook di Workspace berbagi cluster EMR yang sama untuk menjalankan perintah. Anda dapat melampirkan Workspace ke kluster EMR Amazon yang berjalan di Amazon EC2 atau ke EMR Amazon di klaster virtual EKS dan titik akhir terkelola.

- Ruang kerja dapat beralih ke Availability Zone lain yang Anda kaitkan dengan subnet Studio. Anda dapat menghentikan dan memulai ulang Workspace untuk meminta proses failover. Saat memulai ulang Workspace, EMR Studio meluncurkan Workspace di Availability Zone yang berbeda di VPC Studio saat Studio dikonfigurasi dengan akses ke beberapa Availability Zone. Jika Studio hanya memiliki satu Availability Zone, EMR Studio mencoba meluncurkan Workspace di subnet yang berbeda. Untuk informasi selengkapnya, lihat Mengatasi masalah konektivitas Workspace.
- Workspace dapat terhubung ke cluster di salah satu subnet yang terkait dengan Studio.

Untuk informasi selengkapnya tentang membuat dan mengonfigurasi Workspace EMR Studio, lihat Pelajari dasar-dasar Ruang Kerja.

Penyimpanan notebook di Amazon EMR Studio

Saat Anda menggunakan Workspace, EMR Studio menyimpan otomatis sel dalam file notebook dengan irama reguler di lokasi Amazon S3 yang terkait dengan Studio Anda. Proses pencadangan ini mempertahankan pekerjaan antar sesi sehingga Anda dapat kembali ke sana nanti tanpa melakukan perubahan pada repositori Git. Untuk informasi selengkapnya, lihat Menyimpan konten Workspace.

Ketika Anda menghapus file notebook dari Workspace, EMR Studio menghapus versi pencadangan dari Amazon S3 untuk Anda. Namun, jika Anda menghapus Workspace tanpa terlebih dahulu menghapus file notebook, file notebook tetap berada di Amazon S3 dan terus bertambah biaya penyimpanan. Untuk mempelajari informasi lebih lanjut, lihat Menghapus file Workspace dan notebook.

Pertimbangan EMR Studio

Pertimbangan-pertimbangan

Pertimbangkan hal berikut ketika Anda bekerja dengan EMR Studio:

- EMR Studio tersedia sebagai berikut: Wilayah AWS
  - `af-south-1`— Afrika (Cape Town)
  - `ap-east-1`— Asia Pasifik (Hong Kong)
  - `ap-northeast-1`— Asia Pasifik (Tokyo)
  - `ap-northeast-2`- Asia Pasifik (Seoul)
  - `ap-northeast-3`— Asia Pasifik (Osaka) *

- `ap-south-1`— Asia Pasifik (Mumbai)
- `ap-southeast-1`— Asia Pasifik (Singapura)
- `ap-southeast-2`— Asia Pasifik (Sydney)
- `ap-southeast-3`— Asia Pasifik (Jakarta) *
- `ca-central-1`— Kanada (Tengah)
- `eu-central-1`— Eropa (Frankfurt)
- `eu-north-1`— Eropa (Stockholm)
- `eu-west-1`— Eropa (Irlandia)
- `eu-west-2`— Eropa (London)
- `eu-west-3`- Eropa (Paris)
- `eu-south-1`— Eropa (Milan)
- `me-south-1`Timur Tengah (Bahrain)
- `me-central-1`— Timur Tengah (UEA) *
- `sa-east-1`— Amerika Selatan (São Paulo)
- `us-east-1`- AS Timur (Virginia N.)
- `us-east-2`— AS Timur (Ohio)
- `us-west-1`— AS Barat (California N.)
- `us-west-2`— AS Barat (Oregon)

  * UI Spark tidak didukung di Wilayah ini.

- Agar pengguna dapat menyediakan kluster EMR baru yang berjalan di Amazon EC2 untuk Workspace, Anda dapat mengaitkan EMR Studio dengan sekumpulan templat klaster. Administrator dapat menentukan template cluster dengan Service Catalog dan dapat memilih apakah pengguna atau grup dapat mengakses template cluster, atau tidak ada template cluster, dalam Studio.
- Saat Anda menentukan izin akses ke file notebook yang disimpan di Amazon S3 atau membaca rahasia, gunakan AWS Secrets Manager peran layanan Amazon EMR. Kebijakan sesi tidak didukung dengan izin ini.
- Anda dapat membuat beberapa EMR Studios untuk mengontrol akses ke cluster EMR di VPC yang berbeda.

- Gunakan AWS CLI untuk mengatur Amazon EMR di kluster EKS. Anda kemudian dapat menggunakan antarmuka Studio untuk melampirkan cluster ke Workspaces dengan endpoint terkelola untuk menjalankan pekerjaan notebook.
- Ada pertimbangan tambahan ketika Anda menggunakan propagasi identitas tepercaya dengan Amazon EMR yang juga berlaku untuk EMR Studio. Untuk informasi selengkapnya, lihat Pertimbangan dan batasan untuk Amazon EMR dengan integrasi Pusat Identitas.
- EMR Studio tidak mendukung perintah ajaib Python berikut:
  - `%alias`
  - `%alias_magic`
  - `%automagic`
  - `%macro`
  - `%%js`
  - `%%javascript`
  - Memodifikasi `proxy_user` menggunakan `%configure`
  - Memodifikasi `KERNEL_USERNAME` menggunakan `%env` atau `%set_env`
- Amazon EMR di kluster EKS tidak mendukung perintah SparkMagic untuk EMR Studio.
- Untuk menulis pernyataan Scala multi-baris di sel notebook, pastikan bahwa semua kecuali baris terakhir berakhir dengan titik. Contoh berikut menggunakan sintaks yang benar untuk pernyataan Scala multi-baris.

```
val df = spark.sql("SELECT * from table_name).
        filter("col1=='value'").
        limit(50)
```

- Untuk meningkatkan keamanan aplikasi off-console yang mungkin Anda gunakan dengan Amazon EMR, domain hosting aplikasi terdaftar di Daftar Akhiran Publik (PSL). Contoh domain hosting ini meliputi:`emrstudio-prod.us-east-1.amazonaws.com`,`emrnotebooks-prod.us-east-1.amazonaws.com`,`emrappui-prod.us-east-1.amazonaws.com`. Untuk keamanan lebih lanjut, jika Anda perlu mengatur cookie sensitif di nama domain default, kami sarankan Anda menggunakan cookie dengan `__Host-` awalan. Ini membantu mempertahankan domain Anda dari upaya pemalsuan permintaan lintas situs (CSRF). Untuk informasi selengkapnya, lihat Set-Cookiehalaman di Jaringan Pengembang Mozilla.

Masalah yang diketahui

- Studio EMR yang menggunakan Pusat Identitas IAM dengan propagasi identitas tepercaya diaktifkan hanya dapat dikaitkan dengan kluster EMR yang juga menggunakan propagasi identitas tepercaya.
- Pastikan Anda menonaktifkan alat manajemen proxy seperti FoxyProxy atau SwitchyOmega di browser sebelum membuat Studio. Proksi aktif dapat menyebabkan kesalahan saat Anda memilih Buat Studio, dan menghasilkan pesan galat Kegagalan Jaringan.
- Kernel yang berjalan di Amazon EMR di kluster EKS dapat gagal dimulai karena masalah batas waktu. Jika Anda mengalami kesalahan atau masalah saat memulai kernel, tutup file notebook, matikan kernel, lalu buka kembali file notebook.
- Operasi kernel Restart tidak berfungsi seperti yang diharapkan saat Anda menggunakan EMR Amazon di kluster EKS. Setelah Anda memilih Restart kernel, segarkan Workspace agar restart diterapkan.
- Jika Workspace tidak dilampirkan ke klaster, pesan kesalahan akan muncul saat pengguna Studio membuka file notebook dan mencoba memilih kernel. Anda dapat mengabaikan pesan kesalahan ini dengan memilih Oke, tetapi Anda harus melampirkan Workspace ke klaster dan memilih kernel agar Anda dapat menjalankan kode notebook.
- Saat Anda menggunakan Amazon EMR 6.2.0 dengan konfigurasi keamanan untuk mengatur keamanan klaster, antarmuka Workspace tampak kosong dan tidak berfungsi seperti yang diharapkan. Kami menyarankan Anda menggunakan versi Amazon EMR yang didukung berbeda jika Anda ingin mengonfigurasi enkripsi data atau otorisasi Amazon S3 untuk EMRFS untuk klaster. EMR Studio bekerja dengan Amazon EMR versi 5.32.0 (Amazon EMR 5.x series) dan 6.2.0 (Amazon EMR 6.x series) dan lebih tinggi.
- Saat Anda Men-debug Amazon EMR yang berjalan pada pekerjaan Amazon EC2, tautan ke Spark UI pada klaster mungkin tidak bekerja atau gagal untuk muncul. Untuk meregenerasi tautan, buat sel notebook baru dan jalankan perintah `%%info`.
- Jupyter Enterprise Gateway tidak membersihkan kernel idle pada node utama cluster dalam versi rilis Amazon EMR berikut: 5.32.0, 5.33.0, 6.2.0, dan 6.3.0. Kernel idle mengkonsumsi sumber daya komputasi dan dapat menyebabkan cluster yang berjalan lama gagal. Anda dapat mengonfigurasi pembersihan kernel idle untuk Jupyter Enterprise Gateway menggunakan contoh skrip berikut. Anda dapat Connect ke node utama menggunakan SSH, atau mengirimkan skrip sebagai langkah. Untuk informasi selengkapnya, lihat Menjalankan perintah dan skrip di klaster EMR Amazon.

```
#!/bin/bash
```

```
sudo tee -a /emr/notebook-env/conf/jupyter_enterprise_gateway_config.py << EOF
c.MappingKernelManager.cull_connected = True
c.MappingKernelManager.cull_idle_timeout = 10800
c.MappingKernelManager.cull_interval = 300
EOF
sudo systemctl daemon-reload
sudo systemctl restart jupyter_enterprise_gateway
```

- Saat Anda menggunakan kebijakan penghentian otomatis dengan Amazon EMR versi 5.32.0, 5.33.0, 6.2.0, atau 6.3.0, Amazon EMR menandai klaster sebagai idle dan dapat menghentikan klaster secara otomatis meskipun Anda memiliki kernel Python3 yang aktif. Ini karena menjalankan kernel Python3 tidak mengirimkan pekerjaan Spark di cluster. Untuk menggunakan penghentian otomatis dengan kernel Python3, sebaiknya gunakan Amazon EMR versi 6.4.0 atau yang lebih baru. Untuk informasi selengkapnya tentang penghentian otomatis, lihatMenggunakan kebijakan penghentian otomatis.
- Saat Anda menggunakan `%%display` untuk menampilkan Spark DataFrame dalam tabel, tabel yang sangat lebar mungkin terpotong. Anda dapat mengklik kanan output dan memilih Buat Tampilan Baru untuk Output untuk mendapatkan tampilan output yang dapat digulir.
- Memulai kernel berbasis Spark, seperti, Spark PySpark, atau SparkR, memulai sesi Spark, dan menjalankan sel di notebook mengantri pekerjaan Spark di sesi itu. Saat Anda mengganggu sel yang sedang berjalan, pekerjaan Spark terus berjalan. Untuk menghentikan pekerjaan Spark, Anda harus menggunakan UI Spark on-cluster. Untuk petunjuk tentang cara menyambung ke UI Spark, lihatDebug aplikasi dan pekerjaan dengan EMR Studio.

Batasan fitur

Amazon EMR Studio tidak mendukung fitur Amazon EMR berikut:

- Melampirkan dan menjalankan pekerjaan pada cluster EMR dengan konfigurasi keamanan yang menentukan otentikasi Kerberos
- Cluster dengan beberapa node primer
- Cluster yang menggunakan instans Amazon EC2 berdasarkan AWS Graviton2 untuk Amazon EMR 6.x rilis lebih rendah dari 6.9.0, dan rilis 5.x lebih rendah dari 5.36.1

Fitur berikut tidak didukung dari Studio yang menggunakan propagasi identitas tepercaya:

- Membuat cluster EMR tanpa template.

- Menggunakan aplikasi EMR Tanpa Server.
- Meluncurkan Amazon EMR di kluster EKS.
- Menggunakan peran runtime.
- Mengaktifkan kolaborasi SQL Explorer atau Workspace.

Kuota layanan untuk EMR Studio

Tabel berikut menampilkan batas layanan untuk EMR Studio.

Item	Kuota
EMR Studio	Maksimal 100 per AWS akun
Subnet	Maksimum 5 yang terkait dengan setiap EMR Studio
Grup Pusat Identitas IAM	Maksimum 5 yang ditetapkan untuk setiap EMR Studio
Pengguna Pusat Identitas IAM	Maksimum 100 yang ditetapkan untuk setiap EMR Studio

Praktik terbaik VPC dan subnet

Gunakan praktik terbaik berikut untuk menyiapkan Amazon Virtual Private Cloud (Amazon VPC) dengan subnet untuk EMR Studio:

- Anda dapat menentukan maksimal lima subnet di VPC Anda untuk dikaitkan dengan Studio. Kami menyarankan Anda menyediakan beberapa subnet di Availability Zone yang berbeda untuk mendukung ketersediaan Workspace dan memberi pengguna Studio akses ke cluster di berbagai Availability Zone. Untuk mempelajari lebih lanjut tentang bekerja dengan VPC, subnet, dan Availability Zone, lihat VPC dan subnet di Panduan Pengguna. Amazon Virtual Private Cloud
- Subnet yang Anda tentukan harus dapat berkomunikasi satu sama lain.
- Untuk memungkinkan pengguna menautkan Workspace ke repositori Git yang dihosting publik, Anda harus menentukan hanya subnet pribadi yang memiliki akses ke internet melalui Network Address Translation (NAT). Untuk informasi selengkapnya tentang menyiapkan subnet pribadi untuk Amazon EMR, lihat. Subnet privat

- Saat Anda menggunakan Amazon EMR di EKS dengan EMR Studio, setidaknya harus ada satu subnet yang sama antara Studio Anda dan klaster Amazon EKS yang Anda gunakan untuk mendaftarkan cluster virtual. Jika tidak, endpoint terkelola Anda tidak akan muncul sebagai opsi di Studio Workspaces. Anda dapat membuat klaster Amazon EKS dan mengaitkannya dengan subnet milik Studio, atau membuat Studio dan menentukan subnet kluster EKS Anda.
- Jika Anda berencana untuk menggunakan Amazon Amazon EMR di EKS dengan EMR Studio, pilih VPC yang sama dengan node pekerja klaster Amazon EKS Anda.

Persyaratan klaster untuk Amazon EMR Studio

Cluster EMR Amazon Berjalan di Amazon EC2

Semua klaster Amazon EMR yang berjalan di Amazon EC2 yang Anda buat untuk EMR Studio Workspace harus memenuhi persyaratan berikut. Cluster yang Anda buat menggunakan antarmuka EMR Studio secara otomatis memenuhi persyaratan ini.

- Cluster harus menggunakan Amazon EMR versi 5.32.0 (Amazon EMR 5.x series) atau 6.2.0 (Amazon EMR 6.x series) atau yang lebih baru. Anda dapat membuat klaster menggunakan konsol Amazon EMR, atau SDKAWS Command Line Interface, lalu melampirkannya ke EMR Studio Workspace. Pengguna studio juga dapat menyediakan dan melampirkan cluster saat membuat atau bekerja di Amazon EMR Workspace. Untuk informasi selengkapnya, lihat Lampirkan komputasi ke Ruang Kerja EMR Studio.
- Cluster harus berada dalam Amazon Virtual Private Cloud. Platform EC2-Classic tidak didukung.
- Cluster harus menginstal Spark, Livy, dan Jupyter Enterprise Gateway. Jika Anda berencana untuk menggunakan cluster untuk SQL Explorer, Anda harus menginstal Presto dan Spark.
- Untuk menggunakan SQL Explorer, cluster harus menggunakan Amazon EMR versi 5.34.0 atau yang lebih baru atau versi 6.4.0 atau yang lebih baru dan memiliki Presto diinstal. Jika Anda ingin menentukan Katalog Data AWS Glue sebagai metastore Hive untuk Presto, Anda harus mengkonfigurasinya di cluster. Untuk informasi selengkapnya, lihat Menggunakan Presto dengan Katalog Glue Data AWS.
- Cluster harus berada dalam subnet pribadi dengan terjemahan alamat jaringan (NAT) untuk menggunakan repositori Git yang dihosting publik dengan EMR Studio.

Kami merekomendasikan konfigurasi cluster berikut saat Anda bekerja dengan EMR Studio.

- Setel mode penerapan untuk sesi Spark ke mode cluster. Mode cluster menempatkan proses master aplikasi pada node inti dan bukan pada node utama cluster. Melakukannya mengurangi simpul utama dari tekanan memori potensial. Untuk informasi selengkapnya, lihat Gambaran Umum Mode Cluster di dokumentasi Apache Spark.
- Ubah batas waktu Livy dari default satu jam menjadi enam jam seperti pada konfigurasi contoh berikut.

```
{
    "classification":"livy-conf",
        "Properties":{
            "livy.server.session.timeout":"6h",
            "livy.spark.deploy-mode":"cluster"
        }
}
```

- Buat armada instans yang beragam dengan hingga 30 instans, dan pilih beberapa jenis instans di armada Instans Spot Anda. Misalnya, Anda dapat menentukan jenis instance yang dioptimalkan memori berikut untuk beban kerja Spark: r5.2x, r5.4x, r5.8x, r5.12x, r5.16x, r4.2x, r4.4x, r4.8x, r4.12, dll. Untuk informasi selengkapnya, lihat Mengkonfigurasi armada instans.
- Gunakan strategi alokasi yang dioptimalkan kapasitas untuk Instans Spot untuk membantu Amazon EMR membuat pilihan instans yang efektif berdasarkan wawasan kapasitas real-time dari Amazon EC2. Untuk informasi selengkapnya, lihat Strategi alokasi untuk armada instans.
- Aktifkan penskalaan terkelola di klaster Anda. Tetapkan parameter node inti maksimum ke kapasitas persisten minimum yang Anda rencanakan untuk digunakan, dan konfigurasikan penskalaan pada armada tugas yang terdiversifikasi dengan baik yang berjalan di Instans Spot untuk menghemat biaya. Untuk informasi selengkapnya, lihat Menggunakan penskalaan terkelola di Amazon EMR.

Kami juga mendorong Anda untuk menjaga Amazon EMR Block Public Access diaktifkan, dan itu untuk membatasi lalu lintas SSH masuk ke sumber tepercaya. Akses masuk ke klaster memungkinkan pengguna menjalankan notebook pada klaster. Untuk informasi lebih lanjut, lihat Menggunakan Akses publik blok Amazon EMR dan Mengendalikan lalu lintas jaringan dengan grup keamanan.

Amazon EMR di Kluster EKS

Selain klaster EMR yang berjalan di Amazon EC2, Anda dapat menyiapkan dan mengelola Amazon EMR pada klaster EKS untuk EMR Studio menggunakan AWS CLI. Siapkan Amazon EMR di kluster EKS menggunakan pedoman berikut:

- Buat titik akhir HTTPS terkelola untuk EMR Amazon di kluster EKS. Pengguna melampirkan Workspace ke endpoint terkelola. Cluster Amazon Elastic Kubernetes Service (EKS) yang Anda gunakan untuk mendaftarkan klaster virtual harus memiliki subnet pribadi untuk mendukung endpoint terkelola.
- Gunakan klaster Amazon EKS dengan setidaknya satu subnet pribadi dan terjemahan alamat jaringan (NAT) saat Anda ingin menggunakan repositori Git yang dihosting publik.
- Hindari penggunaan AMI Linux Arm Amazon Amazon Amazon yang dioptimalkan Amazon EKS, yang tidak didukung untuk Amazon EMR pada titik akhir yang dikelola EKS.
- Hindari menggunakan kluster Amazon EKS AWS Fargate -only, yang tidak didukung.

Konfigurasikan Amazon EMR Studio

Bagian ini untuk administrator EMR Studio. Ini mencakup cara menyiapkan EMR Studio untuk tim Anda dan memberikan instruksi untuk tugas-tugas seperti menetapkan pengguna dan grup, menyiapkan template cluster, dan mengoptimalkan Apache Spark untuk EMR Studio.

Topik

- Izin administrator untuk membuat dan mengelola EMR Studio
- Menyiapkan Amazon EMR Studio
- Mengelola Amazon EMR Studio
- Menentukan grup keamanan untuk mengontrol lalu lintas jaringan EMR Studio
- Buat AWS CloudFormation template untuk Amazon EMR Studio
- Membuat akses dan izin untuk repositori berbasis Git
- Optimalkan lowongan kerja Spark di EMR Studio

Izin administrator untuk membuat dan mengelola EMR Studio

Izin IAM yang dijelaskan di halaman ini memungkinkan Anda untuk membuat dan mengelola Studio EMR. Untuk informasi mendetail tentang tiap izin yang diperlukan, lihat Izin yang diperlukan untuk mengelola EMR Studio.

Izin yang diperlukan untuk mengelola EMR Studio

Tabel berikut mencantumkan operasi yang terkait dengan membuat dan mengelola EMR Studio. Tabel juga menampilkan izin yang diperlukan untuk setiap operasi.

> **Note**
>
> Anda hanya memerlukan `SessionMapping` tindakan Pusat Identitas IAM dan Studio saat Anda menggunakan mode autentikasi Pusat Identitas IAM.

Izin untuk membuat dan mengelola EMR Studio

Operasi	Izin
Membuat Studio	`"elasticmapreduce:CreateStudio", "sso:CreateApplication", "sso:PutApplicationAuthenticationMethod", "sso:PutApplicationGrant", "sso:PutApplicationAccessScope", "sso:PutApplicationAssignmentConfiguration", "iam:PassRole"`
Menjelaskan Studio	`"elasticmapreduce:DescribeStudio", "sso:GetManagedApplicationInstance"`
Daftar Studios	`"elasticmapreduce:ListStudios"`
Menghapus Studio	`"elasticmapreduce:DeleteStudio", "sso:DeleteApplication", "sso:DeleteApplicationAuthenticationMethod", "sso:DeleteApplicationAccessScope", "sso:DeleteApplicationGrant"`
Additional permissions required when you use IAM Identity Center mode	

Operasi	Izin
Menetapkan pengguna atau grup ke Studio	`"elasticmapreduce:CreateStudioSessionMapping", "sso:GetProfile", "sso:ListDirectoryAssociations", "sso:ListProfiles", "sso:AssociateProfile", "sso-directory:SearchUsers", "sso-directory:SearchGroups", "sso-directory:DescribeUser", "sso-directory:DescribeGroup", "sso:ListInstances", "sso:CreateApplicationAssignment", "sso:DescribeInstance", "organizations:DescribeOrganization", "organizations:ListDelegatedAdministrators", "sso:CreateInstance", "sso:DescribeRegisteredRegions", "sso:GetSharedSsoConfiguration", "iam:ListPolicies"`
Ambil detail tugas Studio untuk pengguna atau grup tertentu	`"sso-directory:SearchUsers", "sso-directory:SearchGroups", "sso-directory:DescribeUser", "sso-directory:DescribeGroup", "sso:DescribeApplication", "elasticmapreduce:GetStudioSessionMapping"`
Mencantumkan semua pengguna dan grup yang ditetapkan ke Studio	`"elasticmapreduce:ListStudioSessionMappings"`

Operasi	Izin
Memperbarui kebijakan sesi yang dilampirkan ke pengguna atau grup yang ditetapkan ke Studio	`"sso-directory:SearchUsers", "sso-directory:SearchGroups", "sso-directory:DescribeUser", "sso-directory:DescribeGroup", "sso:DescribeApplication", "sso:DescribeInstance", "elasticmapreduce:UpdateStudioSessionMapping"`
Menghapus pengguna atau grup dari Studio	`"elasticmapreduce:DeleteStudioSessionMapping", "sso-directory:SearchUsers", "sso-directory:SearchGroups", "sso-directory:DescribeUser", "sso-directory:DescribeGroup", "sso:ListDirectoryAssociations", "sso:GetProfile", "sso:DescribeApplication", "sso:DescribeInstance", "sso:ListProfiles", "sso:DisassociateProfile", "sso:DeleteApplicationAssignment", "sso:ListApplicationAssignments"`

Untuk membuat kebijakan dengan izin admin untuk EMR Studio

1. Ikuti petunjuk dalam Membuat kebijakan IAM untuk membuat kebijakan menggunakan salah satu contoh berikut. Izin yang Anda butuhkan bergantung pada mode otentikasi Anda untuk EMR Studio.

   Masukkan nilai Anda sendiri untuk item ini:

   - Ganti *<your-resource-ARN>* untuk menentukan Amazon Resource Name (ARN) objek atau objek yang menyertakan pernyataan untuk kasus penggunaan Anda.
   - Ganti <region>dengan kode Wilayah AWS tempat Anda berencana membuat Studio.
   - Ganti <aws-account_id>dengan ID AWS akun untuk Studio.

- Ganti <EMRStudio-Service-Role>dan <EMRStudio-User-Role>dengan nama peran layanan EMR Studio dan peran pengguna EMR Studio Anda.

Example Contoh kebijakan: Izin admin saat Anda menggunakan mode autentikasi IAM

```
{
    "Version": "2012-10-17",
    "Statement": [
        {
            "Effect": "Allow",
            "Resource": "arn:aws:elasticmapreduce:<region>:<aws-account-id>:studio/
*",
            "Action": [
                "elasticmapreduce:CreateStudio",
                "elasticmapreduce:DescribeStudio",
                "elasticmapreduce:DeleteStudio"
            ]
        },
        {
            "Effect": "Allow",
            "Resource": "<your-resource-ARN>",
            "Action": [
                "elasticmapreduce:ListStudios"
            ]
        },
        {
            "Effect": "Allow",
            "Resource": [
                "arn:aws:iam::<aws-account-id>:role/<EMRStudio-Service-Role>"
            ],
            "Action": "iam:PassRole"
        }
    ]
}
```

Example Contoh kebijakan: Izin admin saat Anda menggunakan mode autentikasi Pusat Identitas IAM

> **Note**
>
> API direktori Pusat Identitas dan Pusat Identitas tidak mendukung penetapan ARN dalam elemen sumber daya pernyataan kebijakan IAM. Untuk mengizinkan akses ke IAM Identity Center dan IAM Identity Center Directory, izin berikut menentukan semua sumber daya, “Resource” :"*”, untuk tindakan IAM Identity Center. Untuk informasi selengkapnya, lihat Kunci tindakan, sumber daya, dan kondisi untuk Direktori Pusat Identitas IAM.

```
{
    "Version": "2012-10-17",
    "Statement": [
        {
            "Effect": "Allow",
            "Resource": "arn:aws:elasticmapreduce:<region>:<aws-account-id>:studio/
*",
            "Action": [
                "elasticmapreduce:CreateStudio",
                "elasticmapreduce:DescribeStudio",
                "elasticmapreduce:DeleteStudio",
                "elasticmapreduce:CreateStudioSessionMapping",
                "elasticmapreduce:GetStudioSessionMapping",
                "elasticmapreduce:UpdateStudioSessionMapping",
                "elasticmapreduce:DeleteStudioSessionMapping"
            ]
        },
        {
            "Effect": "Allow",
            "Resource": "<your-resource-ARN>",
            "Action": [
                "elasticmapreduce:ListStudios",
                "elasticmapreduce:ListStudioSessionMappings"
            ]
        },
        {
            "Effect": "Allow",
```

```
            "Resource": [
                "arn:aws:iam::<aws-account-id>:role/<EMRStudio-Service-Role>",
                "arn:aws:iam::<aws-account-id>:role/<EMRStudio-User-Role>"
            ],
            "Action": "iam:PassRole"
        },
        {
            "Effect": "Allow",
            "Resource": "*",
            "Action": [
                "sso:CreateApplication",
                "sso:PutApplicationAuthenticationMethod",
                "sso:PutApplicationGrant",
                "sso:PutApplicationAccessScope",
                "sso:PutApplicationAssignmentConfiguration",
                "sso:DescribeApplication",
                "sso:DeleteApplication",
                "sso:DeleteApplicationAuthenticationMethod",
                "sso:DeleteApplicationAccessScope",
                "sso:DeleteApplicationGrant",
                "sso:ListInstances",
                "sso:CreateApplicationAssignment",
                "sso:DeleteApplicationAssignment",
                "sso:ListApplicationAssignments",
                "sso:DescribeInstance",
                "sso:AssociateProfile",
                "sso:DisassociateProfile",
                "sso:GetProfile",
                "sso:ListDirectoryAssociations",
                "sso:ListProfiles",
                "sso-directory:SearchUsers",
                "sso-directory:SearchGroups",
                "sso-directory:DescribeUser",
                "sso-directory:DescribeGroup",
                "organizations:DescribeOrganization",
                "organizations:ListDelegatedAdministrators",
                "sso:CreateInstance",
                "sso:DescribeRegisteredRegions",
                "sso:GetSharedSsoConfiguration",
                "iam:ListPolicies"
            ]
        }
    ]
```

```
}
```

2. Lampirkan kebijakan ke identitas IAM Anda (pengguna, peran, atau grup). Untuk instruksinya, lihat Menambahkan dan menghapus izin identitas IAM.

Menyiapkan Amazon EMR Studio

Selesaikan langkah-langkah berikut untuk menyiapkan Amazon EMR Studio.

Sebelum Anda mulai

> **Note**
>
> Jika Anda berencana untuk menggunakan EMR Studio dengan Amazon EMR di EKS, kami sarankan Anda terlebih dahulu menyiapkan Amazon EMR di EKS untuk EMR Studio sebelum Anda menyiapkan Studio.

Sebelum Anda mengatur EMR Studio, pastikan Anda memiliki item berikut:

- Sesi Akun AWS. Untuk petunjuk, silakan lihat Mengatur Amazon EMR.
- Izin untuk membuat dan mengelola EMR Studio. Untuk informasi selengkapnya, lihat the section called “Izin administrator untuk membuat EMR Studio”.
- Bucket Amazon S3 tempat EMR Studio dapat mencadangkan Workspace dan file notebook di Studio Anda. Untuk petunjuk, lihat Membuat bucket di Panduan Pengguna Amazon Simple Storage Service (S3).
- Jika Anda ingin melampirkan ke EMR Amazon di EC2 atau Amazon EMR di kluster EKS, atau menggunakan repositori Git, Anda memerlukan Amazon Virtual Private Cloud (VPC) untuk Studio, dan maksimal lima subnet. Anda tidak memerlukan VPC untuk menggunakan EMR Studio dengan EMR Tanpa Server. Untuk tips tentang cara mengkonfigurasi jaringan, lihatPraktik terbaik VPC dan subnet.

Untuk menyiapkan EMR Studio

1. Pilih mode otentikasi untuk Amazon EMR Studio
2. Buat sumber daya Studio berikut.
   - Membuat peran layanan EMR Studio

- Konfigurasikan izin pengguna EMR Studio untuk Amazon EC2 atau Amazon EKS
- (Opsional)Menentukan grup keamanan untuk mengontrol lalu lintas jaringan EMR Studio.

3. Membuat EMR Studio
4. Menetapkan pengguna atau grup ke EMR Studio

Setelah Anda menyelesaikan langkah-langkah pengaturan, Anda bisaMenggunakan Amazon EMR Studio.

Pilih mode otentikasi untuk Amazon EMR Studio

EMR Studio mendukung dua mode otentikasi: mode otentikasi IAM dan mode otentikasi IAM Identity Center. Mode IAM menggunakan AWS Identity and Access Management (IAM), sedangkan mode IAM Identity Center menggunakan. AWS IAM Identity Center Saat Anda membuat EMR Studio, Anda memilih mode otentikasi untuk semua pengguna Studio tersebut. Untuk informasi selengkapnya tentang mode otentikasi yang berbeda, lihatOtentikasi dan login pengguna.

Gunakan tabel berikut untuk memilih mode otentikasi untuk EMR Studio.

Jika Anda...	Kami merekomendasikan...
Sudah akrab dengan atau sebelumnya telah mengatur otentikasi atau federasi IAM	Mode otentikasi IAM, yang menawarkan manfaat sebagai berikut: • Menyediakan pengaturan cepat untuk EMR Studio jika Anda sudah mengelola identitas seperti pengguna dan grup di IAM. • Bekerja dengan penyedia identitas yang kompatibel dengan OpenID Connect (OIDC) atau Security Assertion Markup Language 2.0 (SAMP 2.0). • Mendukung penggunaan beberapa penyedia identitas dengan hal yang samaAkun AWS. • Tersedia dalam jumlah yang luasWilayah AWS. • Sesuai dengan SOC 2.

Jika Anda...	Kami merekomendasikan...
Baru di AWS atau Amazon EMR	Mode otentikasi Pusat Identitas IAM, yang menyediakan fitur-fitur berikut: • Mendukung penugasan pengguna dan grup yang mudah ke AWS sumber daya. • Bekerja dengan Microsoft Active Directory dan penyedia identitas SAMP 2.0. • Memfasilitasi pengaturan federasi multi-aku n sehingga Anda tidak perlu mengonfigurasi federasi secara terpisah untuk masing-ma sing Akun AWS di organisasi Anda.

Mengatur mode otentikasi IAM untuk Amazon EMR Studio

Dengan mode otentikasi IAM, Anda dapat menggunakan otentikasi IAM atau federasi IAM. Autentikasi IAM memungkinkan Anda mengelola identitas IAM seperti pengguna, grup, dan peran di IAM. Anda memberi pengguna akses ke Studio dengan kebijakan izin IAM dan kontrol akses berbasis atribut (ABAC). Federasi IAM memungkinkan Anda membangun kepercayaan antara penyedia identitas pihak ketiga (iDP) AWS dan sehingga Anda dapat mengelola identitas pengguna melalui IDP Anda.

> **Note**
>
> Jika Anda sudah menggunakan IAM untuk mengontrol akses ke AWS sumber daya, atau jika Anda sudah mengonfigurasi penyedia identitas (iDP) untuk IAM, Izin pengguna untuk mode otentikasi IAM lihat untuk mengatur izin pengguna saat Anda menggunakan mode autentikasi IAM untuk EMR Studio.

Gunakan federasi IAM untuk Amazon EMR Studio

Untuk menggunakan federasi IAM untuk EMR Studio, Anda membuat hubungan kepercayaan antara penyedia identitas Akun AWS Anda dan penyedia identitas Anda (IDP) dan memungkinkan pengguna federasi untuk mengakses. AWS Management Console Langkah-langkah yang Anda ambil untuk menciptakan hubungan kepercayaan ini berbeda tergantung pada standar federasi IDP Anda.

Secara umum, Anda menyelesaikan tugas-tugas berikut untuk mengkonfigurasi federasi dengan iDP eksternal. Untuk petunjuk selengkapnya, lihat Mengaktifkan pengguna federasi SAMP 2.0 untuk mengakses AWS Management Console dan Mengaktifkan akses broker identitas kustom ke AWS Management Console dalam Panduan Pengguna. AWS Identity and Access Management

1. Kumpulkan informasi dari IDP Anda. Ini biasanya berarti menghasilkan dokumen metadata untuk memvalidasi permintaan otentikasi SAMP dari IDP Anda.
2. Buat entitas IAM penyedia identitas untuk menyimpan informasi tentang IDP Anda. Untuk petunjuk, lihat Membuat penyedia identitas IAM.
3. Buat satu atau beberapa peran IAM untuk IDP Anda. EMR Studio memberikan peran ke pengguna federasi saat pengguna masuk. Peran ini memungkinkan IDP Anda untuk meminta kredenal keamanan sementara untuk akses ke. AWS Untuk petunjuknya, lihat Membuat peran untuk penyedia identitas pihak ketiga (federasi). Kebijakan izin yang Anda tetapkan ke peran menentukan apa yang dapat dilakukan pengguna federasi di dalam AWS dan di Studio EMR. Untuk informasi selengkapnya, lihat Izin pengguna untuk mode otentikasi IAM.
4. (Untuk penyedia SAMP) Lengkapi kepercayaan SAMP dengan mengonfigurasi IDP Anda dengan informasi tentang AWS dan peran yang Anda inginkan untuk diasumsikan oleh pengguna federasi. Proses konfigurasi ini menciptakan kepercayaan pihak yang mengandalkan antara AWS IDP Anda dan. Untuk informasi selengkapnya, lihat Mengonfigurasi IDP SAMP 2.0 Anda dengan mengandalkan kepercayaan pihak dan menambahkan klaim.

Untuk mengkonfigurasi EMR Studio sebagai aplikasi SAMP di portal iDP Anda

Anda dapat mengonfigurasi EMR Studio tertentu sebagai aplikasi SAMP menggunakan deep link ke Studio. Melakukannya memungkinkan pengguna masuk ke portal iDP Anda dan meluncurkan Studio tertentu alih-alih menavigasi melalui konsol EMR Amazon.

- Gunakan format berikut untuk mengonfigurasi deep link ke EMR Studio Anda sebagai URL pendaratan setelah verifikasi pernyataan SAMP.

```
https://console.aws.amazon.com/emr/home?region=<aws-region>#studio/<your-studio-id>/start
```

Mengatur mode otentikasi Pusat Identitas IAM untuk Amazon EMR Studio

AWS IAM Identity CenterUntuk mempersiapkan EMR Studio, Anda harus mengonfigurasi sumber identitas dan menyediakan pengguna dan grup. Provisioning adalah proses membuat informasi

pengguna dan grup tersedia untuk digunakan oleh IAM Identity Center dan oleh aplikasi yang menggunakan IAM Identity Center. Untuk informasi lebih lanjut, lihat Penyediaan pengguna dan grup.

EMR Studio mendukung penggunaan penyedia identitas berikut untuk IAM Identity Center:

- AWS Managed Microsoft AD dan Direktori Aktif yang dikelola sendiri — Untuk informasi lebih lanjut, lihat Menghubungkan ke direktori Microsoft AD.
- Penyedia berbasis SAML – Untuk daftar lengkap, lihat Penyedia identitas yang didukung.
- Direktori Pusat Identitas IAM — Untuk informasi selengkapnya, lihat Mengelola identitas di Pusat Identitas IAM.

Untuk mengatur Pusat Identitas IAM untuk EMR Studio

1. Untuk menyiapkan Pusat Identitas IAM untuk EMR Studio, Anda memerlukan yang berikut ini:
   - Akun manajemen di organisasi AWS Anda jika menggunakan beberapa akun di organisasi.

     > **Note**
     >
     > Anda hanya boleh menggunakan akun manajemen Anda untuk mengaktifkan Pusat Identitas IAM dan menyediakan pengguna dan grup. Setelah menyiapkan Pusat Identitas IAM, gunakan akun anggota untuk membuat EMR Studio dan menetapkan pengguna dan grup. Untuk mempelajari selengkapnya tentang terminologi AWS, lihat terminologi dan konsep AWS Organizations.

   - Jika Anda mengaktifkan Pusat Identitas IAM sebelum 25 November 2019, Anda mungkin harus mengaktifkan aplikasi yang menggunakan Pusat Identitas IAM untuk akun di organisasi AndaAWS. Untuk informasi selengkapnya, lihat Mengaktifkan aplikasi terintegrasi Pusat Identitas IAM di akun. AWS
   - Pastikan Anda memiliki prasyarat yang tercantum di halaman prasyarat Pusat Identitas IAM.
2. Ikuti petunjuk di Aktifkan Pusat Identitas IAM untuk mengaktifkan Pusat Identitas IAM di Wilayah AWS tempat Anda ingin membuat EMR Studio.
3. Connect IAM Identity Center ke penyedia identitas Anda dan berikan pengguna dan grup yang ingin Anda tetapkan ke Studio.

Jika Anda menggunakan...	Lakukan ini...
Direktori Microsoft AD	1. Ikuti instruksi di Menghubungkan ke direktori Microsoft AD untuk menghubungkan Direktori Aktif yang dikelola sendiri atau direktori AWS Managed Microsoft AD menggunakan AWS Directory Service. 2. Untuk menyediakan pengguna dan grup untuk Pusat Identitas IAM, Anda dapat menyinkronkan data identitas dari AD sumber Anda ke Pusat Identitas IAM. Anda dapat menyinkronkan identitas dari iklan sumber Anda dengan berbagai cara. Salah satu caranya adalah dengan menetapkan pengguna atau grup AD ke AWS akun di organisasi Anda. Untuk instruksi, lihat Single sign-on. Sinkronisasi bisa memakan waktu hingga dua jam. Setelah Anda menyelesaikan langkah ini, pengguna dan grup yang disinkronkan muncul di Toko Identitas Anda. **Note** Pengguna dan grup tidak muncul di Toko Identitas Anda sampai Anda menyinkronkan informasi pengguna dan grup atau menggunakan penyediaan pengguna just-in-time (JIT). Untuk informasi lebih lanjut, lihat Penyediaan saat pengguna berasal dari Direktori Aktif.

Jika Anda menggunakan...	Lakukan ini...
	3. (Opsional) Setelah Anda menyinkronkan pengguna dan grup AD, Anda dapat menghapus akses mereka ke AWS akun yang Anda konfigurasikan pada langkah sebelumnya. Untuk instruksi, lihat Menghapus akses pengguna.
Penyedia identitas eksternal	Ikuti instruksi di Menghubungkan ke penyedia identitas eksternal.
Direktori Pusat Identitas IAM	Saat Anda membuat pengguna dan grup di Pusat Identitas IAM, penyediaan dilakukan secara otomatis. Untuk informasi selengkap nya, lihat Mengelola identitas di Pusat Identitas IAM.

Sekarang Anda dapat menetapkan pengguna dan grup dari Identity Store Anda ke EMR Studio. Untuk petunjuk, silakan lihat Menetapkan pengguna atau grup ke EMR Studio.

Membuat peran layanan EMR Studio

Tentang peran layanan EMR Studio

Setiap EMR Studio menggunakan peran IAM dengan izin yang memungkinkan Studio berinteraksi dengan layanan lain. AWS Peran layanan ini harus menyertakan izin yang memungkinkan EMR Studio membuat saluran jaringan aman antara Workspaces dan cluster, menyimpan file Amazon S3 Control notebook, dan mengakses saat menautkan Workspace ke AWS Secrets Manager repositori Git.

Gunakan peran layanan Studio (bukan kebijakan sesi) untuk menentukan semua izin akses Amazon S3 untuk menyimpan file notebook, dan untuk menentukan AWS Secrets Manager izin akses.

Cara membuat peran layanan untuk EMR Studio di Amazon EC2 atau Amazon EKS

1. Ikuti petunjuk dalam Membuat peran untuk mendelegasikan izin ke AWS layanan guna membuat peran layanan dengan kebijakan kepercayaan berikut.

> **Important**
>
> Kebijakan kepercayaan berikut mencakup kunci kondisi `aws:SourceAccount` global `aws:SourceArn` dan untuk membatasi izin yang Anda berikan kepada EMR Studio ke sumber daya tertentu di akun Anda. Melakukannya dapat melindungi Anda dari masalah wakil yang membingungkan.

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Effect": "Allow",
      "Principal": {
        "Service": "elasticmapreduce.amazonaws.com"
      },
      "Action": "sts:AssumeRole",
      "Condition": {
        "StringEquals": {
          "aws:SourceAccount": "<account-id>"
        },
        "ArnLike": {
          "aws:SourceArn": "arn:aws:elasticmapreduce:<region>:<account-id>:*"
        }
      }
    }
  ]
}
```

2. Hapus izin peran default. Kemudian, sertakan izin dari contoh kebijakan izin IAM berikut. Atau, Anda dapat membuat kebijakan khusus yang menggunakanIzin peran layanan EMR Studio.

> **Important**
>
> - Agar kontrol akses berbasis tag Amazon EC2 berfungsi dengan EMR Studio, Anda harus menyetel akses untuk `ModifyNetworkInterfaceAttribute` API seperti yang ditunjukkan kebijakan berikut.
> - Agar EMR Studio bekerja dengan peran layanan, Anda tidak boleh mengubah pernyataan berikut:

`AllowAddingEMRTagsDuringDefaultSecurityGroupCreation` dan. `AllowAddingTagsDuringEC2ENICreation`

- Untuk menggunakan kebijakan contoh, Anda harus menandai sumber daya berikut dengan kunci "**`for-use-with-amazon-emr-managed-policies`**" dan nilai"**`true`**".
  - Amazon Virtual Private Cloud (VPC) Anda untuk EMR Studio.
  - Setiap subnet yang ingin Anda gunakan dengan Studio.
  - Setiap grup keamanan EMR Studio kustom. Anda harus menandai grup keamanan apa pun yang Anda buat selama periode pratinjau EMR Studio jika Anda ingin terus menggunakannya.
  - Rahasia yang disimpan di Studio AWS Secrets Manager yang digunakan pengguna untuk menautkan repositori Git ke Workspace.

  Anda dapat menerapkan tag ke sumber daya menggunakan tab Tag pada layar sumber daya yang relevan di AWS Management Console.

Jika berlaku, ubah kebijakan `* "Resource":"*"` dalam kebijakan berikut untuk menentukan Nama Sumber Daya Amazon (ARN) sumber daya yang dicakup pernyataan tersebut untuk kasus penggunaan Anda.

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Sid": "AllowEMRReadOnlyActions",
      "Effect": "Allow",
      "Action": [
        "elasticmapreduce:ListInstances",
        "elasticmapreduce:DescribeCluster",
        "elasticmapreduce:ListSteps"
      ],
      "Resource": "*"
    },
    {
      "Sid": "AllowEC2ENIActionsWithEMRTags",
      "Effect": "Allow",
      "Action": [
        "ec2:CreateNetworkInterfacePermission",
```

```
        "ec2:DeleteNetworkInterface"
      ],
      "Resource": [
        "arn:aws:ec2:*:*:network-interface/*"
      ],
      "Condition": {
        "StringEquals": {
          "aws:ResourceTag/for-use-with-amazon-emr-managed-policies": "true"
        }
      }
    },
    {
      "Sid": "AllowEC2ENIAttributeAction",
      "Effect": "Allow",
      "Action": [
        "ec2:ModifyNetworkInterfaceAttribute"
      ],
      "Resource": [
        "arn:aws:ec2:*:*:instance/*",
        "arn:aws:ec2:*:*:network-interface/*",
        "arn:aws:ec2:*:*:security-group/*"
      ]
    },
    {
      "Sid": "AllowEC2SecurityGroupActionsWithEMRTags",
      "Effect": "Allow",
      "Action": [
        "ec2:AuthorizeSecurityGroupEgress",
        "ec2:AuthorizeSecurityGroupIngress",
        "ec2:RevokeSecurityGroupEgress",
        "ec2:RevokeSecurityGroupIngress",
        "ec2:DeleteNetworkInterfacePermission"
      ],
      "Resource": "*",
      "Condition": {
        "StringEquals": {
          "aws:ResourceTag/for-use-with-amazon-emr-managed-policies": "true"
        }
      }
    },
    {
      "Sid": "AllowDefaultEC2SecurityGroupsCreationWithEMRTags",
      "Effect": "Allow",
      "Action": [
```

```
        "ec2:CreateSecurityGroup"
      ],
      "Resource": [
        "arn:aws:ec2:*:*:security-group/*"
      ],
      "Condition": {
        "StringEquals": {
          "aws:RequestTag/for-use-with-amazon-emr-managed-policies": "true"
        }
      }
    },
    {
      "Sid": "AllowDefaultEC2SecurityGroupsCreationInVPCWithEMRTags",
      "Effect": "Allow",
      "Action": [
        "ec2:CreateSecurityGroup"
      ],
      "Resource": [
        "arn:aws:ec2:*:*:vpc/*"
      ],
      "Condition": {
        "StringEquals": {
          "aws:ResourceTag/for-use-with-amazon-emr-managed-policies": "true"
        }
      }
    },
    {
      "Sid": "AllowAddingEMRTagsDuringDefaultSecurityGroupCreation",
      "Effect": "Allow",
      "Action": [
        "ec2:CreateTags"
      ],
      "Resource": "arn:aws:ec2:*:*:security-group/*",
      "Condition": {
        "StringEquals": {
          "aws:RequestTag/for-use-with-amazon-emr-managed-policies": "true",
          "ec2:CreateAction": "CreateSecurityGroup"
        }
      }
    },
    {
      "Sid": "AllowEC2ENICreationWithEMRTags",
      "Effect": "Allow",
      "Action": [
```

```
        "ec2:CreateNetworkInterface"
      ],
      "Resource": [
        "arn:aws:ec2:*:*:network-interface/*"
      ],
      "Condition": {
        "StringEquals": {
          "aws:RequestTag/for-use-with-amazon-emr-managed-policies": "true"
        }
      }
    },
    {
      "Sid": "AllowEC2ENICreationInSubnetAndSecurityGroupWithEMRTags",
      "Effect": "Allow",
      "Action": [
        "ec2:CreateNetworkInterface"
      ],
      "Resource": [
        "arn:aws:ec2:*:*:subnet/*",
        "arn:aws:ec2:*:*:security-group/*"
      ],
      "Condition": {
        "StringEquals": {
          "aws:ResourceTag/for-use-with-amazon-emr-managed-policies": "true"
        }
      }
    },
    {
      "Sid": "AllowAddingTagsDuringEC2ENICreation",
      "Effect": "Allow",
      "Action": [
        "ec2:CreateTags"
      ],
      "Resource": "arn:aws:ec2:*:*:network-interface/*",
      "Condition": {
        "StringEquals": {
          "ec2:CreateAction": "CreateNetworkInterface"
        }
      }
    },
    {
      "Sid": "AllowEC2ReadOnlyActions",
      "Effect": "Allow",
      "Action": [
```

```
        "ec2:DescribeSecurityGroups",
        "ec2:DescribeNetworkInterfaces",
        "ec2:DescribeTags",
        "ec2:DescribeInstances",
        "ec2:DescribeSubnets",
        "ec2:DescribeVpcs"
      ],
      "Resource": "*"
    },
    {
      "Sid": "AllowSecretsManagerReadOnlyActionsWithEMRTags",
      "Effect": "Allow",
      "Action": [
        "secretsmanager:GetSecretValue"
      ],
      "Resource": "arn:aws:secretsmanager:*:*:secret:*",
      "Condition": {
        "StringEquals": {
          "aws:ResourceTag/for-use-with-amazon-emr-managed-policies": "true"
        }
      }
    },
    {
      "Sid": "AllowWorkspaceCollaboration",
      "Effect": "Allow",
      "Action": [
        "iam:GetUser",
        "iam:GetRole",
        "iam:ListUsers",
        "iam:ListRoles",
        "sso:GetManagedApplicationInstance",
        "sso-directory:SearchUsers"
      ],
      "Resource": "*"
    }
  ]
}
```

3. Berikan akses baca dan tulis peran layanan Anda ke lokasi Amazon S3 Anda untuk EMR Studio. Gunakan set minimum izin berikut. Untuk informasi lebih lanjut, lihat contoh Amazon S3: Memungkinkan akses baca dan tulis ke objek dalam Bucket S3, secara terprogram dan di konsol.

```
"s3:PutObject",
"s3:GetObject",
"s3:GetEncryptionConfiguration",
"s3:ListBucket",
"s3:DeleteObject"
```

Jika Anda mengenkripsi bucket Amazon S3, sertakan izin berikut untuk. AWS Key Management Service

```
"kms:Decrypt",
"kms:GenerateDataKey",
"kms:ReEncryptFrom",
"kms:ReEncryptTo",
"kms:DescribeKey"
```

4. Jika Anda ingin mengontrol akses ke rahasia Git di tingkat pengguna, tambahkan izin berbasis tag ke **`secretsmanager:GetSecretValue`** dalam kebijakan peran pengguna EMR Studio, dan hapus izin ke kebijakan **`secretsmanager:GetSecretValue`** dari kebijakan peran layanan EMR Studio. Untuk informasi selengkapnya tentang menyetel izin pengguna berbutir halus, lihat. Membuat kebijakan izin untuk pengguna EMR Studio

Peran layanan minimum untuk EMR Tanpa Server

Jika Anda ingin menjalankan beban kerja interaktif dengan EMR Tanpa Server melalui buku catatan EMR Studio, gunakan kebijakan kepercayaan yang sama yang Anda gunakan untuk menyiapkan EMR Studio di bagian sebelumnya. Cara membuat peran layanan untuk EMR Studio di Amazon EC2 atau Amazon EKS

Untuk kebijakan IAM Anda, kebijakan minimum yang layak memiliki izin sebagai berikut. Perbarui *`bucket-name`* dengan nama bucket yang akan Anda gunakan saat mengonfigurasi EMR Studio dan Workspace. EMR Studio menggunakan bucket untuk mencadangkan file Workspaces dan notebook di Studio Anda.

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

```
            "Action": [
               "s3:PutObject",
               "s3:GetObject",
               "s3:DeleteObject"
            ],
            "Resource": ["arn:aws:s3:::bucket-name/*"]
        },
        {
            "Sid": "BucketActions",
            "Effect": "Allow",
            "Action": [
               "s3:ListBucket",
               "s3:GetEncryptionConfiguration"
            ],
            "Resource": ["arn:aws:s3:::bucket-name"]
        }
    ]
}
```

Jika Anda berencana menggunakan bucket Amazon S3 terenkripsi, tambahkan izin berikut pada kebijakan Anda:

```
"kms:Decrypt",
"kms:GenerateDataKey",
"kms:ReEncryptFrom",
"kms:ReEncryptTo",
"kms:DescribeKey"
```

Izin peran layanan EMR Studio

Tabel berikut mencantumkan operasi yang dilakukan EMR Studio menggunakan peran layanan, bersama dengan tindakan IAM yang diperlukan untuk setiap operasi.

Operasi	Tindakan

Menetapkan saluran jaringan aman antara Workspace dan klaster EMR, serta melakukan tindakan pembersihan yang diperlukan.	`"ec2:CreateNetworkInterface",`
	`"ec2:CreateNetworkInterfacePermission",`
	`"ec2:DeleteNetworkInterface",`
	`"ec2:DeleteNetworkInterfacePermission",`
	`"ec2:DescribeNetworkInterfaces",`
	`"ec2:ModifyNetworkInterfaceAttribute",`

Operasi	Tindakan
	"ec2:AuthorizeSecurityGroupEgress", "ec2:AuthorizeSecurityGroupIngress", "ec2:CreateSecurityGroup", "ec2:DescribeSecurityGroups", "ec2:RevokeSecurityGroupEgress", "ec2:DescribeTags", "ec2:DescribeInstances", "ec2:DescribeSubnets", "ec2:DescribeVpcs", "elasticmapreduce:ListInstances", "elasticmapreduce:DescribeCluster", "elasticmapreduce:ListSteps"
Menggunakan kredensial Git yang disimpan di AWS Secrets Manager untuk menautkan repositori Git ke Workspace.	"secretsmanager:GetSecretValue"
Menerapkan tag AWS ke antarmuka jaringan dan grup keamanan default yang dibuat EMR Studio saat menyiapkan saluran jaringan aman. Untuk informasi lebih lanjut, lihat Menandai sumber daya AWS.	"ec2:CreateTags"

Operasi	Tindakan

Mengakses atau mengunggah file notebook dan metadata ke Amazon S3.	`"s3:PutObject",`
	`"s3:GetObject",`
	`"s3:GetEncryptionConfiguration",`
	`"s3:ListBucket",`
	`"s3:DeleteObject"`
	
	Jika Anda menggunakan bucket Amazon S3 terenkripsi, sertakan izin berikut.
	
	`"kms:Decrypt",`
	`"kms:GenerateDataKey",`
	`"kms:ReEncryptFrom",`
	`"kms:ReEncryptTo",`
	`"kms:DescribeKey"`

Aktifkan dan konfigurasikan kolaborasi Workspace.	`"iam:GetUser",`
	`"iam:GetRole",`
	`"iam:ListUsers",`
	`"iam:ListRoles",`
	`"sso:GetManagedApplicationInstance",`
	`"sso-directory:SearchUsers"`

Konfigurasikan izin pengguna EMR Studio untuk Amazon EC2 atau Amazon EKS

Anda harus mengonfigurasi kebijakan izin pengguna untuk Amazon EMR Studio sehingga Anda dapat menyetel izin pengguna dan grup yang berbutir halus. Untuk informasi tentang cara kerja izin pengguna di EMR Studio, Kontrol akses lihat di. Cara Kerja Amazon EMR Studio

> **Note**
>
> Izin yang tercakup dalam bagian ini tidak memberlakukan kontrol akses data. Untuk mengelola akses ke set data input, Anda harus mengonfigurasi izin untuk klaster yang digunakan Studio Anda. Untuk informasi selengkapnya, lihat Keamanan di Amazon EMR.

Buat peran pengguna EMR Studio untuk mode autentikasi IAM Identity Center

Anda harus membuat peran pengguna EMR Studio saat menggunakan mode autentikasi Pusat Identitas IAM.

Untuk membuat peran pengguna untuk EMR Studio

1. Ikuti petunjuk dalam Membuat peran untuk mendelegasikan izin ke AWS layanan di Panduan AWS Identity and Access Management Pengguna untuk membuat peran pengguna.

   Saat Anda membuat peran, gunakan kebijakan hubungan kepercayaan berikut.

   ```
   {
     "Version": "2008-10-17",
     "Statement": [
       {
         "Effect": "Allow",
         "Principal": {
           "Service": "elasticmapreduce.amazonaws.com"
         },
         "Action": [
                 "sts:AssumeRole",
                 "sts:SetContext"
                 ]
       }
     ]
   }
   ```

2. Hapus izin dan kebijakan peran default.
3. Sebelum menetapkan pengguna dan grup ke Studio, lampirkan kebijakan sesi EMR Studio Anda ke peran pengguna. Untuk petunjuk tentang cara membuat kebijakan sesi, lihatMembuat kebijakan izin untuk pengguna EMR Studio.

Membuat kebijakan izin untuk pengguna EMR Studio

Lihat bagian berikut untuk membuat kebijakan izin untuk EMR Studio.

Topik

- Buat kebijakan izin
- Tetapkan kepemilikan untuk kolaborasi Workspace

- Buat kebijakan rahasia Git tingkat pengguna
- Lampirkan kebijakan izin ke identitas IAM Anda

> **Note**
>
> Untuk menyetel izin akses Amazon S3 untuk menyimpan file notebook, dan untuk mengatur izin AWS Secrets Manager akses untuk membaca rahasia saat Anda menautkan Workspaces ke repositori Git, gunakan peran layanan EMR Studio.

Buat kebijakan izin

Buat satu atau beberapa kebijakan izin IAM yang menentukan tindakan apa yang dapat dilakukan pengguna di Studio Anda. Misalnya, Anda dapat membuat tiga kebijakan terpisah untuk tipe pengguna Studio dasar, menengah, dan lanjutan dengan contoh kebijakan di halaman ini.

Untuk rincian setiap operasi Studio yang mungkin dilakukan pengguna, dan tindakan IAM minimum yang diperlukan untuk melakukan setiap operasi, lihatAWS Identity and Access Managementizin untuk pengguna EMR Studio. Untuk langkah-langkah untuk membuat kebijakan, lihat Membuat kebijakan IAM di Panduan Pengguna IAM.

Kebijakan izin Anda harus menyertakan pernyataan berikut.

```
{
            "Sid": "AllowAddingTagsOnSecretsWithEMRStudioPrefix",
            "Effect": "Allow",
            "Action": "secretsmanager:TagResource",
            "Resource": "arn:aws:secretsmanager:*:*:secret:emr-studio-*"
},
{
            "Sid": "AllowPassingServiceRoleForWorkspaceCreation",
            "Action": "iam:PassRole",
            "Resource": [
                "arn:aws:iam::*:role/your-emr-studio-service-role"
            ],
            "Effect": "Allow"
}
```

Tetapkan kepemilikan untuk kolaborasi Workspace

Kolaborasi ruang kerja memungkinkan beberapa pengguna bekerja secara bersamaan di Workspace yang sama dan dapat dikonfigurasi dengan panel Kolaborasi di UI Workspace. Untuk melihat dan menggunakan panel Kolaborasi, pengguna harus memiliki izin berikut. Setiap pengguna dengan izin ini dapat melihat dan menggunakan panel Kolaborasi.

```
"elasticmapreduce:UpdateEditor",
"elasticmapreduce:PutWorkspaceAccess",
"elasticmapreduce:DeleteWorkspaceAccess",
"elasticmapreduce:ListWorkspaceAccessIdentities"
```

Untuk membatasi akses ke panel Kolaborasi, Anda dapat menggunakan kontrol akses berbasis tag. Saat pengguna membuat Workspace, EMR Studio menerapkan tag default dengan kunci `creatorUserId` yang nilainya adalah ID pengguna yang membuat Workspace.

> **Note**
>
> EMR Studio menambahkan `creatorUserId` tag ke Ruang Kerja yang dibuat setelah 16 November 2021. Untuk membatasi siapa saja yang dapat mengonfigurasi kolaborasi untuk ruang kerja yang Anda buat sebelum tanggal ini, sebaiknya tambahkan `creatorUserId` tag secara manual ke Ruang Kerja, lalu gunakan kontrol akses berbasis tag dalam kebijakan izin pengguna.

Pernyataan contoh berikut memungkinkan pengguna mengonfigurasi kolaborasi untuk Workspace apa pun dengan kunci tag `creatorUserId` yang nilainya cocok dengan ID pengguna (ditunjukkan oleh variabel kebijakan`aws:userId`). Dengan kata lain, pernyataan tersebut memungkinkan pengguna mengonfigurasi kolaborasi untuk Workspaces yang mereka buat. Untuk mempelajari lebih lanjut tentang variabel kebijakan, lihat elemen kebijakan IAM: Variabel dan tag di Panduan Pengguna IAM.

```
    {
        "Sid": "UserRolePermissionsForCollaboration",
        "Action": [
            "elasticmapreduce:UpdateEditor",
            "elasticmapreduce:PutWorkspaceAccess",
            "elasticmapreduce:DeleteWorkspaceAccess",
            "elasticmapreduce:ListWorkspaceAccessIdentities"
```

```
        ],
        "Resource": "*",
        "Effect": "Allow",
        "Condition": {
            "StringEquals": {
                "elasticmapreduce:ResourceTag/creatorUserId": "${aws:userid}"
            }
        }
    }
```

Buat kebijakan rahasia Git tingkat pengguna

Topik

- Untuk menggunakan izin tingkat pengguna
- Untuk beralih dari izin tingkat layanan ke izin tingkat pengguna
- Untuk menggunakan izin tingkat layanan

Untuk menggunakan izin tingkat pengguna

EMR Studio secara otomatis menambahkan `for-use-with-amazon-emr-managed-user-policies` tag saat membuat rahasia Git. Jika Anda ingin mengontrol akses ke rahasia Git di tingkat pengguna, tambahkan izin berbasis tag ke kebijakan peran pengguna EMR Studio dengan `secretsmanager:GetSecretValue` seperti yang ditunjukkan pada bagian di bawah ini. Untuk beralih dari izin tingkat layanan ke izin tingkat pengguna

Jika Anda memiliki izin yang ada `secretsmanager:GetSecretValue` dalam kebijakan peran layanan EMR Studio, Anda harus menghapus izin tersebut.

Untuk beralih dari izin tingkat layanan ke izin tingkat pengguna

> **Note**
>
> `for-use-with-amazon-emr-managed-user-policies`Tag memastikan bahwa izin dari Langkah 1 di bawah ini memberikan pencipta ruang kerja akses ke rahasia Git. Namun, jika Anda menautkan repositori Git sebelum 1 September 2023, maka rahasia Git yang sesuai akan ditolak aksesnya karena tag tersebut `for-use-with-amazon-emr-managed-user-policies` tidak diterapkan. Untuk menerapkan izin tingkat pengguna, Anda harus membuat ulang rahasia lama dari JupyterLab dan menautkan kembali repositori Git yang sesuai.

Untuk informasi selengkapnya tentang variabel kebijakan, lihat elemen kebijakan IAM: Variabel dan tag di Panduan Pengguna IAM.

1. Tambahkan izin berikut ke kebijakan peran pengguna EMR Studio. Ia menggunakan `for-use-with-amazon-emr-managed-user-policies` kunci dengan nilai`"${aws:userid}"`.

```
{
   "Sid": "AllowSecretsManagerReadOnlyActionsWithEMRTags",
    "Effect": "Allow",
    "Action": "secretsmanager:GetSecretValue",
    "Resource": "arn:aws:secretsmanager:*:*:secret:*",
    "Condition": {
        "StringEquals": {
            "secretsmanager:ResourceTag/for-use-with-amazon-emr-managed-user-
policies": "${aws:userid}"
        }
    }
}
```

2. Jika ada, hapus izin berikut dari kebijakan peran layanan EMR Studio. Karena kebijakan peran layanan berlaku untuk semua rahasia yang ditentukan oleh setiap pengguna, Anda hanya perlu melakukan ini satu kali.

```
{
    "Sid": "AllowSecretsManagerReadOnlyActionsWithEMRTags",
    "Effect": "Allow",
    "Action": [
        "secretsmanager:GetSecretValue"
     ],
    "Resource": "arn:aws:secretsmanager:*:*:secret:*",
    "Condition": {
        "StringEquals": {
            "aws:ResourceTag/for-use-with-amazon-emr-managed-policies": "true"
        }
    }
}
```

Untuk menggunakan izin tingkat layanan

Mulai 1 September 2023, EMR Studio secara otomatis menambahkan tag untuk kontrol `for-use-with-amazon-emr-managed-user-policies` akses tingkat pengguna. Karena ini adalah kemampuan tambahan, Anda dapat terus menggunakan akses tingkat layanan yang tersedia melalui `GetSecretValue` izin dalam peran layanan EMR Studio.

Untuk rahasia yang dibuat sebelum 1 September 2023, EMR Studio tidak menambahkan `for-use-with-amazon-emr-managed-user-policies` tag. Untuk tetap menggunakan izin tingkat layanan, cukup pertahankan peran layanan EMR Studio dan izin peran pengguna yang ada. Namun, untuk membatasi siapa yang dapat mengakses rahasia individu, kami sarankan Anda mengikuti langkah-langkah untuk menambahkan `for-use-with-amazon-emr-managed-user-policies` tag secara manual Untuk menggunakan izin tingkat pengguna ke rahasia Anda, dan kemudian menggunakan kontrol akses berbasis tag dalam kebijakan izin pengguna Anda.

Untuk informasi selengkapnya tentang variabel kebijakan, lihat elemen kebijakan IAM: Variabel dan tag di Panduan Pengguna IAM.

Lampirkan kebijakan izin ke identitas IAM Anda

Tabel berikut merangkum identitas IAM yang Anda lampirkan ke kebijakan izin, tergantung pada mode otentikasi EMR Studio Anda. Untuk petunjuk tentang cara melampirkan kebijakan, lihat Menambahkan dan menghapus izin identitas IAM.

Jika Anda menggunakan...	Lampirkan kebijakan ke...
Autentikasi IAM	Identitas IAM Anda (pengguna, grup pengguna, atau peran). Misalnya, Anda dapat melampirk an kebijakan izin ke pengguna di akun AndaAkun AWS.
Federasi IAM dengan penyedia identitas eksternal (iDP)	Peran atau peran IAM yang Anda buat untuk iDP eksternal Anda. Misalnya, IAM untuk federasi SAMP 2.0. EMR Studio menggunakan izin yang Anda lampirkan ke peran IAM untuk pengguna dengan akses federasi ke Studio.
Pusat Identitas IAM	Peran pengguna Amazon EMR Studio Anda.

Contoh kebijakan pengguna

Kebijakan pengguna dasar berikut memungkinkan sebagian besar tindakan EMR Studio, tetapi tidak mengizinkan pengguna membuat kluster EMR Amazon baru.

Kebijakan dasar

> **Important**
>
> Kebijakan contoh tidak menyertakan `CreateStudioPresignedUrl` izin, yang harus Anda izinkan untuk pengguna saat Anda menggunakan mode autentikasi IAM. Untuk informasi selengkapnya, lihat Menetapkan pengguna atau grup ke EMR Studio.

Kebijakan contoh menyertakan `Condition` elemen untuk menerapkan kontrol akses berbasis tag (TBAC) sehingga Anda dapat menggunakan kebijakan dengan peran layanan contoh untuk EMR Studio. Untuk informasi selengkapnya, lihat Membuat peran layanan EMR Studio.

```
{
   "Version":"2012-10-17",
   "Statement":[
      {
         "Sid":"AllowDefaultEC2SecurityGroupsCreationInVPCWithEMRTags",
         "Effect":"Allow",
         "Action":[
            "ec2:CreateSecurityGroup"
         ],
         "Resource":[
            "arn:aws:ec2:*:*:vpc/*"
         ],
         "Condition":{
            "StringEquals":{
               "aws:ResourceTag/for-use-with-amazon-emr-managed-policies":"true"
            }
         }
      },
      {
         "Sid":"AllowAddingEMRTagsDuringDefaultSecurityGroupCreation",
         "Effect":"Allow",
         "Action":[
            "ec2:CreateTags"
         ],
```

```
            "Resource":"arn:aws:ec2:*:*:security-group/*",
            "Condition":{
               "StringEquals":{
                  "aws:RequestTag/for-use-with-amazon-emr-managed-policies":"true",
                  "ec2:CreateAction":"CreateSecurityGroup"
               }
            }
         },
         {
            "Sid":"AllowSecretManagerListSecrets",
            "Action":[
               "secretsmanager:ListSecrets"
            ],
            "Resource":"*",
            "Effect":"Allow"
         },
         {
            "Sid":"AllowSecretCreationWithEMRTagsAndEMRStudioPrefix",
            "Effect":"Allow",
            "Action":"secretsmanager:CreateSecret",
            "Resource":"arn:aws:secretsmanager:*:*:secret:emr-studio-*",
            "Condition":{
               "StringEquals":{
                  "aws:RequestTag/for-use-with-amazon-emr-managed-policies":"true"
               }
            }
         },
         {
            "Sid":"AllowAddingTagsOnSecretsWithEMRStudioPrefix",
            "Effect":"Allow",
            "Action":"secretsmanager:TagResource",
            "Resource":"arn:aws:secretsmanager:*:*:secret:emr-studio-*"
         },
         {
            "Sid":"AllowPassingServiceRoleForWorkspaceCreation",
            "Action":"iam:PassRole",
            "Resource":[
               "arn:aws:iam::*:role/<your-emr-studio-service-role>"
            ],
            "Effect":"Allow"
         },
         {
            "Sid":"AllowS3ListAndLocationPermissions",
            "Action":[
```

```
            "s3:ListAllMyBuckets",
            "s3:ListBucket",
            "s3:GetBucketLocation"
         ],
         "Resource":"arn:aws:s3:::*",
         "Effect":"Allow"
      },
      {
         "Sid":"AllowS3ReadOnlyAccessToLogs",
         "Action":[
            "s3:GetObject"
         ],
         "Resource":[
            "arn:aws:s3:::aws-logs-<aws-account-id>-<region>/elasticmapreduce/*"
         ],
         "Effect":"Allow"
      },
      {
         "Sid":"AllowConfigurationForWorkspaceCollaboration",
         "Action":[
            "elasticmapreduce:UpdateEditor",
            "elasticmapreduce:PutWorkspaceAccess",
            "elasticmapreduce:DeleteWorkspaceAccess",
            "elasticmapreduce:ListWorkspaceAccessIdentities"
         ],
         "Resource":"*",
         "Effect":"Allow",
         "Condition":{
            "StringEquals":{
               "elasticmapreduce:ResourceTag/creatorUserId":"${aws:userId}"
            }
         }
      },
      {
         "Sid":"DescribeNetwork",
         "Effect":"Allow",
         "Action":[
            "ec2:DescribeVpcs",
            "ec2:DescribeSubnets",
            "ec2:DescribeSecurityGroups"
         ],
         "Resource":"*"
      },
      {
```

```
         "Sid":"ListIAMRoles",
         "Effect":"Allow",
         "Action":[
            "iam:ListRoles"
         ],
         "Resource":"*"
      }
   ]
}
```

Kebijakan pengguna perantara berikut memungkinkan sebagian besar tindakan EMR Studio, dan memungkinkan pengguna membuat kluster EMR Amazon baru menggunakan templat klaster.

Kebijakan menengah

> **Important**
>
> Kebijakan contoh tidak menyertakan `CreateStudioPresignedUrl` izin, yang harus Anda izinkan untuk pengguna saat Anda menggunakan mode autentikasi IAM. Untuk informasi selengkapnya, lihat Menetapkan pengguna atau grup ke EMR Studio.

Kebijakan contoh menyertakan `Condition` elemen untuk menerapkan kontrol akses berbasis tag (TBAC) sehingga Anda dapat menggunakan kebijakan dengan peran layanan contoh untuk EMR Studio. Untuk informasi selengkapnya, lihat Membuat peran layanan EMR Studio.

```
{
   "Version":"2012-10-17",
   "Statement":[
      {
         "Sid":"AllowEMRBasicActions",
         "Action":[
            "elasticmapreduce:CreateEditor",
            "elasticmapreduce:DescribeEditor",
            "elasticmapreduce:ListEditors",
            "elasticmapreduce:StartEditor",
            "elasticmapreduce:StopEditor",
            "elasticmapreduce:DeleteEditor",
            "elasticmapreduce:OpenEditorInConsole",
            "elasticmapreduce:AttachEditor",
            "elasticmapreduce:DetachEditor",
            "elasticmapreduce:CreateRepository",
```

```
            "elasticmapreduce:DescribeRepository",
            "elasticmapreduce:DeleteRepository",
            "elasticmapreduce:ListRepositories",
            "elasticmapreduce:LinkRepository",
            "elasticmapreduce:UnlinkRepository",
            "elasticmapreduce:DescribeCluster",
            "elasticmapreduce:ListInstanceGroups",
            "elasticmapreduce:ListBootstrapActions",
            "elasticmapreduce:ListClusters",
            "elasticmapreduce:ListSteps",
            "elasticmapreduce:CreatePersistentAppUI",
            "elasticmapreduce:DescribePersistentAppUI",
            "elasticmapreduce:GetPersistentAppUIPresignedURL",
            "elasticmapreduce:GetOnClusterAppUIPresignedURL"
         ],
         "Resource":"*",
         "Effect":"Allow"
      },
      {
         "Sid":"AllowEMRContainersBasicActions",
         "Action":[
            "emr-containers:DescribeVirtualCluster",
            "emr-containers:ListVirtualClusters",
            "emr-containers:DescribeManagedEndpoint",
            "emr-containers:ListManagedEndpoints",
            "emr-containers:DescribeJobRun",
            "emr-containers:ListJobRuns"
         ],
         "Resource":"*",
         "Effect":"Allow"
      },
      {
          "Sid": "AllowRetrievingManagedEndpointCredentials",
          "Effect": "Allow",
          "Action": [
              "emr-containers:GetManagedEndpointSessionCredentials"
          ],
          "Resource": [
              "arn:aws:emr-containers:<region>:<account-id>:/virtualclusters/<virtual-
cluster-id>/endpoints/<managed-endpoint-id>"
          ],
          "Condition": {
              "StringEquals": {
                  "emr-containers:ExecutionRoleArn": [
```

```
                    "arn:aws:iam::<account-id>:role/<emr-on-eks-execution-role>"
                ]
            }
        }
    },
    {
       "Sid":"AllowSecretManagerListSecrets",
       "Action":[
          "secretsmanager:ListSecrets"
       ],
       "Resource":"*",
       "Effect":"Allow"
    },
    {
       "Sid":"AllowSecretCreationWithEMRTagsAndEMRStudioPrefix",
       "Effect":"Allow",
       "Action":"secretsmanager:CreateSecret",
       "Resource":"arn:aws:secretsmanager:*:*:secret:emr-studio-*",
       "Condition":{
          "StringEquals":{
             "aws:RequestTag/for-use-with-amazon-emr-managed-policies":"true"
          }
       }
    },
    {
       "Sid":"AllowAddingTagsOnSecretsWithEMRStudioPrefix",
       "Effect":"Allow",
       "Action":"secretsmanager:TagResource",
       "Resource":"arn:aws:secretsmanager:*:*:secret:emr-studio-*"
    },
    {
       "Sid":"AllowClusterTemplateRelatedIntermediateActions",
       "Action":[
          "servicecatalog:DescribeProduct",
          "servicecatalog:DescribeProductView",
          "servicecatalog:DescribeProvisioningParameters",
          "servicecatalog:ProvisionProduct",
          "servicecatalog:SearchProducts",
          "servicecatalog:UpdateProvisionedProduct",
          "servicecatalog:ListProvisioningArtifacts",
          "servicecatalog:ListLaunchPaths",
          "servicecatalog:DescribeRecord",
          "cloudformation:DescribeStackResources"
       ],
```

```
            "Resource":"*",
            "Effect":"Allow"
        },
        {
            "Sid":"AllowPassingServiceRoleForWorkspaceCreation",
            "Action":"iam:PassRole",
            "Resource":[
                "arn:aws:iam::*:role/<your-emr-studio-service-role>"
            ],
            "Effect":"Allow"
        },
        {
            "Sid":"AllowS3ListAndLocationPermissions",
            "Action":[
                "s3:ListAllMyBuckets",
                "s3:ListBucket",
                "s3:GetBucketLocation"
            ],
            "Resource":"arn:aws:s3:::*",
            "Effect":"Allow"
        },
        {
            "Sid":"AllowS3ReadOnlyAccessToLogs",
            "Action":[
                "s3:GetObject"
            ],
            "Resource":[
                "arn:aws:s3:::aws-logs-<aws-account-id>-<region>/elasticmapreduce/*"
            ],
            "Effect":"Allow"
        },
        {
            "Sid":"AllowConfigurationForWorkspaceCollaboration",
            "Action":[
                "elasticmapreduce:UpdateEditor",
                "elasticmapreduce:PutWorkspaceAccess",
                "elasticmapreduce:DeleteWorkspaceAccess",
                "elasticmapreduce:ListWorkspaceAccessIdentities"
            ],
            "Resource":"*",
            "Effect":"Allow",
            "Condition":{
                "StringEquals":{
                    "elasticmapreduce:ResourceTag/creatorUserId":"${aws:userId}"
```

```
            }
         }
      },
      {
         "Sid":"DescribeNetwork",
         "Effect":"Allow",
         "Action":[
            "ec2:DescribeVpcs",
            "ec2:DescribeSubnets",
            "ec2:DescribeSecurityGroups"
         ],
         "Resource":"*"
      },
      {
         "Sid":"ListIAMRoles",
         "Effect":"Allow",
         "Action":[
            "iam:ListRoles"
         ],
         "Resource":"*"
      },
      {
         "Sid": "AllowServerlessActions",
         "Action": [
            "emr-serverless:CreateApplication",
            "emr-serverless:UpdateApplication",
            "emr-serverless:DeleteApplication",
            "emr-serverless:ListApplications",
            "emr-serverless:GetApplication",
            "emr-serverless:StartApplication",
            "emr-serverless:StopApplication",
            "emr-serverless:StartJobRun",
            "emr-serverless:CancelJobRun",
            "emr-serverless:ListJobRuns",
            "emr-serverless:GetJobRun",
            "emr-serverless:GetDashboardForJobRun",
            "emr-serverless:AccessInteractiveEndpoints"
         ],
         "Resource": "*",
         "Effect": "Allow"
        },
        {
           "Sid": "AllowPassingRuntimeRoleForRunningServerlessJob",
           "Action": "iam:PassRole",
```

```
            "Resource": "arn:aws:iam::*:role/serverless-runtime-role",
            "Effect": "Allow"
        }
    ]
}
```

Kebijakan pengguna lanjutan berikut memungkinkan semua tindakan EMR Studio, dan memungkinkan pengguna membuat kluster EMR Amazon baru menggunakan templat klaster atau dengan menyediakan konfigurasi cluster.

Kebijakan lanjutan

> **Important**
>
> Kebijakan contoh tidak menyertakan `CreateStudioPresignedUrl` izin, yang harus Anda izinkan untuk pengguna saat Anda menggunakan mode autentikasi IAM. Untuk informasi selengkapnya, lihat Menetapkan pengguna atau grup ke EMR Studio.

Kebijakan contoh menyertakan `Condition` elemen untuk menerapkan kontrol akses berbasis tag (TBAC) sehingga Anda dapat menggunakan kebijakan dengan peran layanan contoh untuk EMR Studio. Untuk informasi selengkapnya, lihat Membuat peran layanan EMR Studio.

```
{
    "Version":"2012-10-17",
    "Statement":[
        {
            "Sid":"AllowEMRBasicActions",
            "Action":[
                "elasticmapreduce:CreateEditor",
                "elasticmapreduce:DescribeEditor",
                "elasticmapreduce:ListEditors",
                "elasticmapreduce:StartEditor",
                "elasticmapreduce:StopEditor",
                "elasticmapreduce:DeleteEditor",
                "elasticmapreduce:OpenEditorInConsole",
                "elasticmapreduce:AttachEditor",
                "elasticmapreduce:DetachEditor",
                "elasticmapreduce:CreateRepository",
                "elasticmapreduce:DescribeRepository",
                "elasticmapreduce:DeleteRepository",
                "elasticmapreduce:ListRepositories",
```

```
            "elasticmapreduce:LinkRepository",
            "elasticmapreduce:UnlinkRepository",
            "elasticmapreduce:DescribeCluster",
            "elasticmapreduce:ListInstanceGroups",
            "elasticmapreduce:ListBootstrapActions",
            "elasticmapreduce:ListClusters",
            "elasticmapreduce:ListSteps",
            "elasticmapreduce:CreatePersistentAppUI",
            "elasticmapreduce:DescribePersistentAppUI",
            "elasticmapreduce:GetPersistentAppUIPresignedURL",
            "elasticmapreduce:GetOnClusterAppUIPresignedURL"
         ],
         "Resource":"*",
         "Effect":"Allow"
      },
      {
         "Sid":"AllowEMRContainersBasicActions",
         "Action":[
            "emr-containers:DescribeVirtualCluster",
            "emr-containers:ListVirtualClusters",
            "emr-containers:DescribeManagedEndpoint",
            "emr-containers:ListManagedEndpoints",
            "emr-containers:DescribeJobRun",
            "emr-containers:ListJobRuns"
         ],
         "Resource":"*",
         "Effect":"Allow"
      },
      {
          "Sid": "AllowRetrievingManagedEndpointCredentials",
          "Effect": "Allow",
          "Action": [
              "emr-containers:GetManagedEndpointSessionCredentials"
          ],
          "Resource": [
              "arn:aws:emr-containers:<region>:<account-id>:/virtualclusters/<virtual-
cluster-id>/endpoints/<managed-endpoint-id>"
          ],
          "Condition": {
              "StringEquals": {
                  "emr-containers:ExecutionRoleArn": [
                      "arn:aws:iam::<account-id>:role/<emr-on-eks-execution-role>"
                  ]
              }
```

```
            }
        },
        {
            "Sid":"AllowSecretManagerListSecrets",
            "Action":[
                "secretsmanager:ListSecrets"
            ],
            "Resource":"*",
            "Effect":"Allow"
        },
        {
            "Sid":"AllowSecretCreationWithEMRTagsAndEMRStudioPrefix",
            "Effect":"Allow",
            "Action":"secretsmanager:CreateSecret",
            "Resource":"arn:aws:secretsmanager:*:*:secret:emr-studio-*",
            "Condition":{
                "StringEquals":{
                    "aws:RequestTag/for-use-with-amazon-emr-managed-policies":"true"
                }
            }
        },
        {
            "Sid":"AllowAddingTagsOnSecretsWithEMRStudioPrefix",
            "Effect":"Allow",
            "Action":"secretsmanager:TagResource",
            "Resource":"arn:aws:secretsmanager:*:*:secret:emr-studio-*"
        },
        {
            "Sid":"AllowClusterTemplateRelatedIntermediateActions",
            "Action":[
                "servicecatalog:DescribeProduct",
                "servicecatalog:DescribeProductView",
                "servicecatalog:DescribeProvisioningParameters",
                "servicecatalog:ProvisionProduct",
                "servicecatalog:SearchProducts",
                "servicecatalog:UpdateProvisionedProduct",
                "servicecatalog:ListProvisioningArtifacts",
                "servicecatalog:ListLaunchPaths",
                "servicecatalog:DescribeRecord",
                "cloudformation:DescribeStackResources"
            ],
            "Resource":"*",
            "Effect":"Allow"
        },
```

```
    {
       "Sid":"AllowEMRCreateClusterAdvancedActions",
       "Action":[
          "elasticmapreduce:RunJobFlow"
       ],
       "Resource":"*",
       "Effect":"Allow"
    },
    {
       "Sid":"AllowPassingServiceRoleForWorkspaceCreation",
       "Action":"iam:PassRole",
       "Resource":[
          "arn:aws:iam::*:role/<your-emr-studio-service-role>",
          "arn:aws:iam::*:role/EMR_DefaultRole_V2",
          "arn:aws:iam::*:role/EMR_EC2_DefaultRole"
       ],
       "Effect":"Allow"
    },
    {
       "Sid":"AllowS3ListAndLocationPermissions",
       "Action":[
          "s3:ListAllMyBuckets",
          "s3:ListBucket",
          "s3:GetBucketLocation"
       ],
       "Resource":"arn:aws:s3:::*",
       "Effect":"Allow"
    },
    {
       "Sid":"AllowS3ReadOnlyAccessToLogs",
       "Action":[
          "s3:GetObject"
       ],
       "Resource":[
          "arn:aws:s3:::aws-logs-<aws-account-id>-<region>/elasticmapreduce/*"
       ],
       "Effect":"Allow"
    },
    {
       "Sid":"AllowConfigurationForWorkspaceCollaboration",
       "Action":[
          "elasticmapreduce:UpdateEditor",
          "elasticmapreduce:PutWorkspaceAccess",
          "elasticmapreduce:DeleteWorkspaceAccess",
```

```
            "elasticmapreduce:ListWorkspaceAccessIdentities"
         ],
         "Resource":"*",
         "Effect":"Allow",
         "Condition":{
            "StringEquals":{
               "elasticmapreduce:ResourceTag/creatorUserId":"${aws:userId}"
            }
         }
      },
      {
         "Sid" : "SageMakerDataWranglerForEMRStudio",
         "Effect" : "Allow",
         "Action" : [
            "sagemaker:CreatePresignedDomainUrl",
            "sagemaker:DescribeDomain",
            "sagemaker:ListDomains",
            "sagemaker:ListUserProfiles"
         ],
         "Resource":"*"
      },
      {
         "Sid":"DescribeNetwork",
         "Effect":"Allow",
         "Action":[
            "ec2:DescribeVpcs",
            "ec2:DescribeSubnets",
            "ec2:DescribeSecurityGroups"
         ],
         "Resource":"*"
      },
      {
         "Sid":"ListIAMRoles",
         "Effect":"Allow",
         "Action":[
            "iam:ListRoles"
         ],
         "Resource":"*"
      },
      {
         "Sid": "AllowServerlessActions",
         "Action": [
            "emr-serverless:CreateApplication",
            "emr-serverless:UpdateApplication",
```

```
            "emr-serverless:DeleteApplication",
            "emr-serverless:ListApplications",
            "emr-serverless:GetApplication",
            "emr-serverless:StartApplication",
            "emr-serverless:StopApplication",
            "emr-serverless:StartJobRun",
            "emr-serverless:CancelJobRun",
            "emr-serverless:ListJobRuns",
            "emr-serverless:GetJobRun",
            "emr-serverless:GetDashboardForJobRun",
            "emr-serverless:AccessInteractiveEndpoints"
        ],
        "Resource": "*",
        "Effect": "Allow"
    },
    {
        "Sid": "AllowPassingRuntimeRoleForRunningServerlessJob",
        "Action": "iam:PassRole",
        "Resource": "arn:aws:iam::*:role/serverless-runtime-role",
        "Effect": "Allow"
    },
    {
        "Sid": "AllowCodeWhisperer",
        "Effect": "Allow",
        "Action": [ "codewhisperer:GenerateRecommendations" ],
        "Resource": "*"
    },
    {
        "Sid": "AllowAthenaSQL",
        "Action": [
            "athena:StartQueryExecution",
            "athena:StopQueryExecution",
            "athena:GetQueryExecution",
            "athena:GetQueryRuntimeStatistics",
            "athena:GetQueryResults",
            "athena:ListQueryExecutions",
            "athena:BatchGetQueryExecution",
            "athena:GetNamedQuery",
            "athena:ListNamedQueries",
            "athena:BatchGetNamedQuery",
            "athena:UpdateNamedQuery",
            "athena:DeleteNamedQuery",
            "athena:ListDataCatalogs",
            "athena:GetDataCatalog",
```

```
            "athena:ListDatabases",
            "athena:GetDatabase",
            "athena:ListTableMetadata",
            "athena:GetTableMetadata",
            "athena:ListWorkGroups",
            "athena:GetWorkGroup",
            "athena:CreateNamedQuery",
            "athena:GetPreparedStatement",
            "glue:CreateDatabase",
            "glue:DeleteDatabase",
            "glue:GetDatabase",
            "glue:GetDatabases",
            "glue:UpdateDatabase",
            "glue:CreateTable",
            "glue:DeleteTable",
            "glue:BatchDeleteTable",
            "glue:UpdateTable",
            "glue:GetTable",
            "glue:GetTables",
            "glue:BatchCreatePartition",
            "glue:CreatePartition",
            "glue:DeletePartition",
            "glue:BatchDeletePartition",
            "glue:UpdatePartition",
            "glue:GetPartition",
            "glue:GetPartitions",
            "glue:BatchGetPartition",
            "kms:ListAliases",
            "kms:ListKeys",
            "kms:DescribeKey",
            "lakeformation:GetDataAccess",
            "s3:GetBucketLocation",
            "s3:GetBucketLocation",
            "s3:GetObject",
            "s3:ListBucket",
            "s3:ListBucketMultipartUploads",
            "s3:ListMultipartUploadParts",
            "s3:AbortMultipartUpload",
            "s3:PutObject",
            "s3:PutBucketPublicAccessBlock",
            "s3:ListAllMyBuckets"
        ],
        "Resource": "*",
        "Effect": "Allow"
```

```
      }
   ]
}
```

Kebijakan pengguna berikut berisi izin pengguna minimum yang diperlukan untuk menggunakan aplikasi interaktif EMR Tanpa Server dengan EMR Studio Workspaces.

EMR Kebijakan interaktif tanpa server

Dalam contoh kebijakan ini yang memiliki izin pengguna untuk aplikasi interaktif EMR Tanpa Server dengan EMR Studio, ganti placeholder *serverless-runtime-role*untuk *emr-studio-service-role*dan dengan peran layanan EMR Studio dan peran runtime EMR Tanpa Server Anda yang benar.

```
{
    "Version": "2012-10-17",
    "Statement": [
        {
            "Sid": "AllowServerlessActions",
            "Action": [
                "emr-serverless:CreateApplication",
                "emr-serverless:UpdateApplication",
                "emr-serverless:DeleteApplication",
                "emr-serverless:ListApplications",
                "emr-serverless:GetApplication",
                "emr-serverless:StartApplication",
                "emr-serverless:StopApplication",
                "emr-serverless:StartJobRun",
                "emr-serverless:CancelJobRun",
                "emr-serverless:ListJobRuns",
                "emr-serverless:GetJobRun",
                "emr-serverless:GetDashboardForJobRun",
                "emr-serverless:AccessInteractiveEndpoints"
            ],
            "Resource": "*",
            "Effect": "Allow"
        },
        {
            "Sid": "AllowEMRBasicActions",
            "Action": [
                "elasticmapreduce:CreateEditor",
                "elasticmapreduce:DescribeEditor",
                "elasticmapreduce:ListEditors",
```

```
                "elasticmapreduce:UpdateStudio",
                "elasticmapreduce:StartEditor",
                "elasticmapreduce:StopEditor",
                "elasticmapreduce:DeleteEditor",
                "elasticmapreduce:OpenEditorInConsole",
                "elasticmapreduce:AttachEditor",
                "elasticmapreduce:DetachEditor",
                "elasticmapreduce:CreateStudio",
                "elasticmapreduce:DescribeStudio",
                "elasticmapreduce:DeleteStudio",
                "elasticmapreduce:ListStudios",
                "elasticmapreduce:CreateStudioPresignedUrl"
            ],
            "Resource": "*",
            "Effect": "Allow"
        },
        {
            "Sid": "AllowPassingRuntimeRoleForRunningEMRServerlessJob",
            "Action": "iam:PassRole",
            "Resource": "arn:aws:iam::*:role/serverless-runtime-role",
            "Effect": "Allow"
        },
        {
            "Sid": "AllowPassingServiceRoleForWorkspaceCreation",
            "Action": "iam:PassRole",
            "Resource": "arn:aws:iam::*:role/emr-studio-service-role",
            "Effect": "Allow"
        },
        {
            "Sid": "AllowS3ListAndGetPermissions",
            "Action": [
                "s3:ListAllMyBuckets",
                "s3:ListBucket",
                "s3:GetBucketLocation",
                "s3:GetObject"
            ],
            "Resource": "arn:aws:s3:::*",
            "Effect": "Allow"
        },
        {
            "Sid":"DescribeNetwork",
            "Effect":"Allow",
            "Action":[
                "ec2:DescribeVpcs",
```

```
                "ec2:DescribeSubnets",
                "ec2:DescribeSecurityGroups"
            ],
            "Resource":"*"
        },
        {
            "Sid": "ListIAMRoles",
            "Effect": "Allow",
            "Action": [
                "iam:ListRoles"
            ],
            "Resource": "*"
        }
    ]
}
```

AWS Identity and Access Managementizin untuk pengguna EMR Studio

Tabel berikut mencakup setiap operasi Amazon EMR Studio yang mungkin dilakukan pengguna, dan mencantumkan tindakan IAM minimum yang diperlukan untuk melakukan operasi tersebut. Anda mengizinkan tindakan ini dalam kebijakan izin IAM (saat Anda menggunakan autentikasi IAM) atau dalam kebijakan sesi peran pengguna (saat Anda menggunakan autentikasi IAM Identity Center) untuk EMR Studio.

Tabel ini juga menampilkan operasi yang diizinkan di setiap contoh kebijakan izin untuk EMR Studio. Untuk informasi selengkapnya tentang contoh kebijakan izin, lihatMembuat kebijakan izin untuk pengguna EMR Studio.

Tindakan	Dasar	Menengah	Advanced	Tindakan terkait
Membuat dan menghapus Workspace	Ya	Ya	Ya	`"elasticmapreduce:CreateEditor", "elasticmapreduce:DescribeEditor", "elasticmapreduce:ListEditors", "elasticmapreduce:DeleteEditor"`
Lihat panel Kolaborasi, aktifkan kolaboras	Ya	Ya	Ya	`"elasticmapreduce:UpdateEditor",`

Tindakan	Dasar	Menengah	Advanced	Tindakan terkait
i Workspace, dan tambahkan kolaborator. Untuk informasi selengkapnya, lihat Menetapkan kepemilikan untuk kolaborasi Workspace.				`"elasticmapreduce:PutWorkspaceAccess", "elasticmapreduce:DeleteWorkspaceAccess", "elasticmapreduce:ListWorkspaceAccessIdentities"`
Lihat daftar bucket Amazon S3 Control penyimpanan di akun yang sama dengan Studio saat membuat klaster EMR baru, dan mengakses log kontainer saat menggunakan UI web untuk men-debug aplikasi	Ya	Ya	Ya	`"s3:ListAllMyBuckets", "s3:ListBucket", "s3:GetBucketLocation", "s3:GetObject"`
Mengakses Workspace	Ya	Ya	Ya	`"elasticmapreduce:DescribeEditor", "elasticmapreduce:ListEditors", "elasticmapreduce:StartEditor", "elasticmapreduce:StopEditor", "elasticmapreduce:OpenEditorInConsole"`

Tindakan	Dasar	Menengah	Advanced	Tindakan terkait
Melampirkan atau melepaskan klaster Amazon EMR yang ada yang terkait dengan Workspace	Ya	Ya	Ya	`"elasticmapreduce:AttachEditor", "elasticmapreduce:DetachEditor", "elasticmapreduce:ListClusters", "elasticmapreduce:DescribeCluster", "elasticmapreduce:ListInstanceGroups", "elasticmapreduce:ListBootstrapActions"`
Melampirkan atau melepaskan Amazon EMR pada klaster EKS	Ya	Ya	Ya	`"elasticmapreduce:AttachEditor", "elasticmapreduce:DetachEditor", "emr-containers:ListVirtualClusters", "emr-containers:DescribeVirtualCluster", "emr-containers:ListManagedEndpoints", "emr-containers:DescribeManagedEndpoint", "emr-containers:GetManagedEndpointSessionCredentials"`

Tindakan	Dasar	Menengah	Advanced	Tindakan terkait
Lampirkan atau lepaskan aplikasi EMR Tanpa Server yang terkait dengan Workspace	Tidak	Ya	Ya	`"elasticmapreduce:AttachEditor", "elasticmapreduce:DetachEditor", "emr-serverless:GetApplication", "emr-serverless:StartApplication", "emr-serverless:ListApplications", "emr-serverless:GetDashboardForJobRun", "emr-serverless:AccessInteractiveEndpoints", "iam:PassRole"` `PassRole`Izin diperlukan untuk lulus peran runtime pekerjaan EMR Tanpa Server. Untuk informasi selengkapnya, lihat Peran runtime Job di Panduan Pengguna Tanpa Server Amazon EMR.

Tindakan	Dasar	Menengah	Advanced	Tindakan terkait
Men-debug pekerjaan Amazon EMR pada EC2 dengan antarmuka pengguna aplikasi persisten	Ya	Ya	Ya	`"elasticmapreduce:CreatePersistentAppUI", "elasticmapreduce:DescribePersistentAppUI", "elasticmapreduce:GetPersistentAppUIPresignedURL", "elasticmapreduce:ListClusters", "elasticmapreduce:ListSteps", "elasticmapreduce:DescribeCluster", "s3:ListBucket", "s3:GetObject"`
Men-debug pekerjaan Amazon EMR pada EC2 dengan antarmuka pengguna aplikasi di klaster	Ya	Ya	Ya	`"elasticmapreduce:GetOnClusterAppUIPresignedURL"`

Tindakan	Dasar	Menengah	Advanced	Tindakan terkait
Men-debug pekerjaan Amazon EMR pada EKS yang berjalan menggunakan Spark History Server	Ya	Ya	Ya	`"elasticmapreduce:CreatePersistentAppUI", "elasticmapreduce:DescribePersistentAppUI", "elasticmapreduce:GetPersistentAppUIPresignedURL", "emr-containers:ListVirtualClusters", "emr-containers:DescribeVirtualCluster", "emr-containers:ListJobRuns", "emr-containers:DescribeJobRun", "s3:ListBucket", "s3:GetObject"`
Membuat dan menghapus repositori Git	Ya	Ya	Ya	`"elasticmapreduce:CreateRepository", "elasticmapreduce:DeleteRepository", "elasticmapreduce:ListRepositories", "elasticmapreduce:DescribeRepository", "secretsmanager:CreateSecret", "secretsmanager:ListSecrets", "secretsmanager:TagResource"`

Tindakan	Dasar	Menengah	Advanced	Tindakan terkait
Menautkan dan menghapus tautan repositori Git	Ya	Ya	Ya	`"elasticmapreduce:LinkRepository", "elasticmapreduce:UnlinkRepository", "elasticmapreduce:ListRepositories", "elasticmapreduce:DescribeRepository"`
Membuat klaster baru dari templat klaster yang telah ditetapkan	Tidak	Ya	Ya	`"servicecatalog:SearchProducts", "servicecatalog:DescribeProduct", "servicecatalog:DescribeProductView", "servicecatalog:DescribeProvisioningParameters", "servicecatalog:ProvisionProduct", "servicecatalog:UpdateProvisionedProduct", "servicecatalog:ListProvisioningArtifacts", "servicecatalog:DescribeRecord", "servicecatalog:ListLaunchPaths", "cloudformation:DescribeStackResources", "elasticmapreduce:ListClusters", "elasticmapreduce:DescribeCluster"`

Tindakan	Dasar	Menengah	Advanced	Tindakan terkait
Menyediakan konfigurasi cluster untuk membuat cluster baru.	Tidak	Tidak	Ya	`"elasticmapreduce:RunJobFlow", "iam:PassRole", "elasticmapreduce:ListClusters", "elasticmapreduce:DescribeCluster"`
Tetapkan pengguna ke Studio saat Anda menggunakan mode autentikasi IAM.	Tidak	Tidak	Tidak	`"elasticmapreduce:CreateStudioPresignedUrl"`
Jelaskan objek jaringan.	Ya	Ya	Ya	`{ "Version": "2012-10-17", "Statement": [ { "Sid": "DescribeNetwork", "Effect": "Allow", "Action": [ "ec2:DescribeVpcs", "ec2:DescribeSubnets", "ec2:DescribeSecurityGroups" ], "Resource": "*" } ] }`

Tindakan	Dasar	Menengah	Advanced	Tindakan terkait
Daftar peran IAM.	Ya	Ya	Ya	`{ "Version": "2012-10-17", "Statement": [ { "Sid": "ListIAMRoles", "Effect": "Allow", "Action": [ "iam:ListRoles" ], "Resource": "*" } ] }`
Connect ke EMR Studio dari Amazon SageMaker Studio dan gunakan antarmuka visual Data Wrangler.	Tidak	Tidak	Ya	`"sagemaker:CreatePresignedDomainUrl", "sagemaker:DescribeDomain", "sagemaker:ListDomains", "sagemaker:ListUserProfiles"`
Gunakan Amazon CodeWhisperer di EMR Studio Anda.	Tidak	Tidak	Ya	`"codewhisperer:GenerateRecommendations"`

Tindakan	Dasar	Menengah	Advanced	Tindakan terkait
Akses editor SQL Amazon Athena dari EMR Studio Anda. Daftar ini mungkin tidak menyertakan semua izin yang Anda perlukan untuk menggunakan semua fitur Athena. Untuk up-to-date daftar terbanyak, lihat kebijakan akses penuh Athena.	Tidak	Tidak	Ya	`"athena:StartQueryExecution", "athena:StopQueryExecution", "athena:GetQueryExecution", "athena:GetQueryRuntimeStatistics", "athena:GetQueryResults", "athena:ListQueryExecutions", "athena:BatchGetQueryExecution", "athena:GetNamedQuery", "athena:ListNamedQueries", "athena:BatchGetNamedQuery", "athena:UpdateNamedQuery", "athena:DeleteNamedQuery", "athena:ListDataCatalogs", "athena:GetDataCatalog", "athena:ListDatabases", "athena:GetDatabase", "athena:ListTableMetadata", "athena:GetTableMetadata", "athena:ListWorkGroups", "athena:GetWorkGroup", "athena:CreateNamedQuery", "athena:GetPreparedStatement", "glue:CreateDatabase", "glue:DeleteDatabase", "glue:GetDatabase", "glue:GetDatabases",`

Tindakan	Dasar	Menengah	Advanced	Tindakan terkait
				"glue:UpdateDatabase", "glue:CreateTable", "glue:DeleteTable", "glue:BatchDeleteTable", "glue:UpdateTable", "glue:GetTable", "glue:GetTables", "glue:BatchCreatePartition", "glue:CreatePartition", "glue:DeletePartition", "glue:BatchDeletePartition", "glue:UpdatePartition", "glue:GetPartition", "glue:GetPartitions", "glue:BatchGetPartition", "kms:ListAliases", "kms:ListKeys", "kms:DescribeKey", "lakeformation:GetDataAccess", "s3:GetBucketLocation", "s3:GetBucketLocation", "s3:GetObject", "s3:ListBucket", "s3:ListBucketMultipartUploads", "s3:ListMultipartUploadParts", "s3:AbortMultipartUpload", "s3:PutObject", "s3:PutBucketPublicAccessBlock", "s3:ListAllMyBuckets"

Membuat EMR Studio

Anda dapat membuat EMR Studio untuk tim Anda dengan konsol Amazon EMR atau. AWS CLI Membuat instance Studio adalah bagian dari pengaturan Amazon EMR Studio.

> **Note**
>
> Kami telah mendesain ulang konsol EMR Amazon agar lebih mudah digunakan. Lihat Apa yang baru dengan konsol? untuk mempelajari tentang perbedaan antara pengalaman konsol lama dan baru.

Prasyarat

Sebelum Anda membuat Studio, pastikan Anda telah menyelesaikan tugas sebelumnyaMenyiapkan Amazon EMR Studio.

Untuk membuat Studio menggunakanAWS CLI, Anda harus menginstal versi terbaru. Untuk informasi selengkapnya, lihat Menginstal atau memperbarui versi terbaru AWS CLI.

> **Important**
>
> Nonaktifkan alat manajemen proxy seperti FoxyProxy atau SwitchyOmega di browser sebelum Anda membuat Studio. Proksi aktif dapat menghasilkan pesan galat Kegagalan Jaringan saat Anda memilih Buat Studio.

New console

Untuk membuat EMR Studio dengan konsol baru

1. Buka konsol EMR Amazon di https://console.aws.amazon.com/emr.
2. Di bawah EMR Studio di navigasi kiri, pilih Memulai. Anda juga dapat membuat Studio baru dari halaman Studios.
3. Pilih Buat Studio untuk membuka halaman Buat Studio.
4. Masukkan Nama Studio dan, secara Deskripsi opsional.
5. Di bawah Autentikasi, pilih mode otentikasi untuk Studio dan berikan informasi sesuai dengan tabel berikut. Untuk mempelajari lebih lanjut tentang otentikasi untuk EMR Studio, lihat. Pilih mode otentikasi untuk Amazon EMR Studio

Jika Anda menggunakan...	Lakukan ini...
Otentikasi atau federasi IAM	Metode otentikasi default adalah AWS Identity and Access Management(IAM). Di bagian bawah layar, Anda juga dapat menambahkan tag untuk memberika n pengguna tertentu akses ke Studio seperti yang dijelaskan dalamMenetapkan pengguna atau grup ke EMR Studio. Jika Anda ingin pengguna federasi masuk menggunakan URL Studio dan kredensio nal untuk penyedia identitas (iDP) Anda, pilih iDP Anda dari daftar tarik-turun, dan masukkan URL login dan nama parameter penyedia Identitas (iDP) Anda. RelayState Untuk daftar URL dan nama autentika si IDP, lihat. RelayState RelayState Parameter penyedia identitas dan URL otentikasi Kemudian, pilih peran EMR Studio Service Anda dari daftar dropdown. Untuk informasi selengkapnya, lihat Membuat peran layanan EMR Studio.

Jika Anda menggunakan...	Lakukan ini...
Autentikasi Pusat Identitas IAM	Pilih Peran Layanan EMR Studio dan Peran Pengguna Anda. Untuk informasi selengkapnya, silakan lihat Membuat peran layanan EMR Studio dan Buat peran pengguna EMR Studio untuk mode autentikasi IAM Identity Center. Saat Anda menggunakan autentikasi IAM Identity Center (sebelumnya AWS Single Sign On) untuk Studio, Anda dapat memilih untuk merampingkan pengalaman masuk bagi pengguna dengan opsi Aktifkan propagasi identitas tepercaya. Dengan propagasi identitas tepercaya, pengguna dapat masuk dengan kredensil Pusat Identitas mereka dan identitas mereka disebarkan ke AWS layanan hilir saat mereka menggunakan Studio. Di bagian Akses aplikasi, Anda juga dapat menentukan apakah semua pengguna dan grup di Pusat Identitas Anda harus memiliki akses ke Studio, atau jika hanya pengguna dan grup yang ditetapkan yang Anda pilih yang dapat mengakses Studio. Untuk informasi selengkapnya, lihatIntegrasi Amazon EMR dengan AWS IAM Identity Center, dan juga propagasi identitas tepercaya di seluruh aplikasi di Panduan Pengguna Pusat AWS Identitas IAM.

6. Di bawah Networking, pilih Amazon Virtual Private Cloud (VPC) untuk Studio dari daftar dropdown.

7. Di bawah Subnet, pilih maksimal lima subnet di VPC Anda untuk dikaitkan dengan Studio. Anda memiliki opsi untuk menambahkan lebih banyak subnet setelah Anda membuat Studio.
8. Untuk grup Keamanan, pilih grup keamanan default atau grup keamanan khusus. Untuk informasi selengkapnya, lihat Menentukan grup keamanan untuk mengontrol lalu lintas jaringan EMR Studio.

Jika Anda memilih...	Lakukan ini...
Grup keamanan EMR Studio default	Untuk mengaktifkan penautan repositor i berbasis Git untuk Studio, pilih Aktifkan kluster/titik akhir dan repositori Git. Jika tidak, pilih Aktifkan klaster/titik akhir.
Grup keamanan khusus untuk Studio Anda	• Di bawah Grup keamanan klaster/t itik akhir, pilih grup keamanan mesin yang telah Anda konfigurasi dari daftar dropdown. Studio Anda menggunakan grup keamanan ini untuk mengizinkan akses masuk dari Workspace terlampir. • Di bawah Grup keamanan Workspace , pilih grup keamanan Workspace yang Anda konfigurasikan dari daftar tarik-turun. Studio Anda menggunakan grup keamanan ini dengan Workspaces untuk menyediakan akses keluar ke klaster EMR Amazon terlampir dan repositori Git yang dihosting secara publik.

9. Di bawah Penyimpanan Workspace, pilih Browse S3 untuk memilih bucket Amazon S3 untuk mencadangkan Workspaces dan file notebook.

> **Note**
>
> Peran layanan EMR Studio Anda harus memiliki akses baca dan tulis ke bucket yang Anda pilih.

10. Pilih Buat Studio untuk menyelesaikan dan arahkan ke halaman Studios. Studio baru Anda muncul dalam daftar dengan detail seperti nama Studio, tanggal pembuatan, dan URL akses Studio.

Setelah Anda membuat Studio, ikuti petunjuk di Menetapkan pengguna atau grup ke EMR Studio.

CLI

> **Note**
>
> Karakter lanjutan baris Linux (\) disertakan agar mudah dibaca Karakter ini bisa dihapus atau digunakan dalam perintah Linux. Untuk Windows, hapus atau ganti dengan tanda sisipan (^).

Example — Buat EMR Studio yang menggunakan IAM untuk otentikasi

Contoh AWS CLI perintah berikut membuat EMR Studio dengan modus otentikasi IAM. Bila Anda menggunakan autentikasi IAM atau federasi untuk Studio, Anda tidak menentukan. `--user-role`

Untuk mengizinkan pengguna federasi masuk menggunakan URL Studio dan kredensional untuk penyedia identitas (iDP) Anda, tentukan dan. `--idp-auth-url` `--idp-relay-state-parameter-name` Untuk daftar URL dan nama autentikasi IDP, lihat. RelayState RelayState Parameter penyedia identitas dan URL otentikasi

```
aws emr create-studio \
--name <example-studio-name> \
--auth-mode IAM \
--vpc-id <example-vpc-id> \
--subnet-ids <subnet-id-1> <subnet-id-2>... <subnet-id-5>  \
--service-role <example-studio-service-role-name> \
--user-role studio-user-role-name \
--workspace-security-group-id <example-workspace-sg-id> \
--engine-security-group-id <example-engine-sg-id> \
--default-s3-location <example-s3-location> \
--idp-auth-url <https://EXAMPLE/login/> \
--idp-relay-state-parameter-name <example-RelayState>
```

Example — Buat Studio EMR yang menggunakan Pusat Identitas untuk otentikasi

AWS CLIContoh perintah berikut membuat EMR Studio yang menggunakan mode autentikasi IAM Identity Center. Bila Anda menggunakan autentikasi IAM Identity Center, Anda harus menentukan. `--user-role`

Untuk informasi selengkapnya tentang mode autentikasi Pusat Identitas IAM, lihat. Mengatur mode otentikasi Pusat Identitas IAM untuk Amazon EMR Studio

```
aws emr create-studio \
--name <example-studio-name> \
--auth-mode SSO \
--vpc-id <example-vpc-id> \
--subnet-ids <subnet-id-1> <subnet-id-2>... <subnet-id-5>  \
--service-role <example-studio-service-role-name> \
--user-role <example-studio-user-role-name> \
--workspace-security-group-id <example-workspace-sg-id> \
--engine-security-group-id <example-engine-sg-id> \
--default-s3-location <example-s3-location>
--trusted-identity-propagation-enabled \
--idc-user-assignment OPTIONAL \
--idc-instance-arn <iam-identity-center-instance-arn>
```

Example — Output CLI untuk **`aws emr create-studio`**

Berikut ini adalah contoh output yang muncul setelah Anda membuat Studio.

```
{
    StudioId: "es-123XXXXXXXXX",
    Url: "https://es-123XXXXXXXXX.emrstudio-prod.us-east-1.amazonaws.com"
}
```

Untuk informasi lebih lanjut tentang perintah `create-studio`, lihat Referensi Perintah AWS CLI.

RelayState Parameter penyedia identitas dan URL otentikasi

Saat Anda menggunakan federasi IAM, dan Anda ingin pengguna masuk menggunakan URL Studio dan kredensil untuk penyedia identitas (iDP), Anda dapat menentukan URL login dan nama parameter penyedia Identitas (iDP) saat Anda. RelayStateMembuat EMR Studio

Tabel berikut menunjukkan URL otentikasi standar dan nama RelayState parameter untuk beberapa penyedia identitas populer.

Penyedia identitas	Parameter	URL otentikasi
Auth0	`RelayState`	`https://<sub_domain> .auth0.com/samlp/<app_id>`
Akun Google	`RelayState`	`https://accounts.google.com/o/saml2/initsso?idpid= <idp_id>&spid=<sp_id>&forceauthn=false`
Microsoft Azure	`RelayState`	`https://myapps.microsoft.com/signin/ <app_name> /<app_id>?tenantId= <tenant_id>`
Okta	`RelayState`	`https://<sub_domain> .okta.com/app/<app_name> /<app_id>/sso/saml`
PingFederate	`TargetResource`	`https://<host>/idp/<idp_id>/startSSO.ping?PartnerSpId= <sp_id>`
PingOne	`TargetResource`	`https://sso.connect.pingidentity.com/sso/sp/initsso?saasid= <app_id>&idpid=<idp_id>`

Menetapkan dan mengelola pengguna EMR Studio

Setelah Anda membuat EMR Studio, Anda dapat menetapkan pengguna dan grup untuk itu. Metode yang Anda gunakan untuk menetapkan, memperbarui, dan menghapus pengguna bergantung pada mode otentikasi Studio.

- Saat Anda menggunakan mode autentikasi IAM, Anda mengonfigurasi penetapan dan izin pengguna EMR Studio di IAM atau dengan IAM dan penyedia identitas Anda.
- Dengan mode autentikasi Pusat Identitas IAM, Anda menggunakan konsol manajemen EMR Amazon atau untuk mengelola pengguna. AWS CLI

Untuk mempelajari lebih lanjut tentang otentikasi Amazon EMR Studio, lihat. Pilih mode otentikasi untuk Amazon EMR Studio

Menetapkan pengguna atau grup ke EMR Studio

IAM

Saat Anda menggunakanMengatur mode otentikasi IAM untuk Amazon EMR Studio, Anda harus mengizinkan `CreateStudioPresignedUrl` tindakan dalam kebijakan izin IAM pengguna dan membatasi pengguna ke Studio tertentu. Anda dapat memasukkan `CreateStudioPresignedUrl` dalam kebijakan Anda Izin pengguna untuk mode otentikasi IAM atau menggunakan kebijakan terpisah.

Untuk membatasi pengguna ke Studio (atau kumpulan Studios), Anda dapat menggunakan kontrol akses berbasis atribut (ABAC) atau menentukan Nama Sumber Daya Amazon (ARN) Studio dalam elemen kebijakan izin. `Resource`

Example Menetapkan pengguna ke Studio menggunakan Studio ARN

Kebijakan contoh berikut memberi pengguna akses ke EMR Studio tertentu dengan mengizinkan `CreateStudioPresignedUrl` tindakan dan menentukan Nama Sumber Daya Amazon (ARN) Studio dalam elemen. `Resource`

```
{
    "Version": "2012-10-17",
    "Statement": [
        {
            "Sid": "AllowCreateStudioPresignedUrl",
            "Effect": "Allow",
            "Action": [
                "elasticmapreduce:CreateStudioPresignedUrl"
            ],
            "Resource": "arn:aws:elasticmapreduce:<region>:<account-
id>:studio/<studio-id>"
        }
    ]
}
```

Example Menetapkan pengguna ke Studio dengan ABAC untuk autentikasi IAM

Ada beberapa cara untuk mengonfigurasi kontrol akses berbasis atribut (ABAC) untuk Studio. Misalnya, Anda dapat melampirkan satu atau beberapa tag ke EMR Studio, lalu membuat

kebijakan IAM yang membatasi `CreateStudioPresignedUrl` tindakan ke Studio atau kumpulan Studio tertentu dengan tag tersebut.

Anda dapat menambahkan tag selama atau setelah pembuatan Studio. Untuk menambahkan tag ke Studio yang ada, Anda dapat menggunakan AWS CLI`emr add-tags`perintah. Contoh berikut menambahkan tag dengan pasangan kunci-nilai `Team = Data Analytics` ke EMR Studio.

```
aws emr add-tags --resource-id <example-studio-id> --tags Team="Data Analytics"
```

Contoh kebijakan izin berikut memungkinkan `CreateStudioPresignedUrl` tindakan untuk EMR Studios dengan pasangan nilai kunci tag. `Team = DataAnalytics` Untuk informasi selengkapnya tentang penggunaan tag untuk mengontrol akses, lihat Mengontrol akses ke dan untuk pengguna dan peran menggunakan tag atau Mengontrol akses ke AWS sumber daya menggunakan tag.

```
{
    "Version": "2012-10-17",
    "Statement": [
        {
            "Sid": "AllowCreateStudioPresignedUrl",
            "Effect": "Allow",
            "Action": [
                "elasticmapreduce:CreateStudioPresignedUrl"
            ],
            "Resource": "arn:aws:elasticmapreduce:<region>:<account-id>:studio/*",
            "Condition": {
                "StringEquals": {
                    "elasticmapreduce:ResourceTag/Team": "Data Analytics"
                }
            }
        }
    ]
}
```

Example Tetapkan pengguna ke Studio menggunakan aws: kunci kondisi SourceIdentity global

Saat Anda menggunakan federasi IAM, Anda dapat menggunakan kunci kondisi global `aws:SourceIdentity` dalam kebijakan izin untuk memberi pengguna akses Studio saat mereka mengambil peran IAM Anda untuk federasi.

Anda harus terlebih dahulu mengonfigurasi penyedia identitas Anda (iDP) untuk mengembalikan string pengidentifikasi, seperti alamat email atau nama pengguna, ketika pengguna mengautentikasi dan mengasumsikan peran IAM Anda untuk federasi. IAM menetapkan kunci kondisi global `aws:SourceIdentity` ke string pengidentifikasi yang dikembalikan oleh idP Anda.

Untuk informasi selengkapnya, lihat Cara menghubungkan aktivitas peran IAM dengan posting blog identitas perusahaan di Blog AWS Keamanan dan aws: SourceIdentity entri dalam referensi kunci kondisi global.

Kebijakan contoh berikut memungkinkan `CreateStudioPresignedUrl` tindakan dan memberi pengguna akses `aws:SourceIdentity` yang cocok dengan *< example-source-identity >* ke EMR Studio yang ditentukan oleh *< example-studio-arn >*.

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Effect": "Allow",
      "Action": "elasticmapreduce:CreateStudioPresignedUrl",
      "Resource": "<example-studio-arn>",
      "Condition": {
        "StringLike": {
          "aws:SourceIdentity": "<example-source-identity>"
        }
      }
    }
  ]
}
```

IAM Identity Center

Saat menetapkan pengguna atau grup ke EMR Studio, Anda menentukan kebijakan sesi yang menentukan izin berbutir halus, seperti kemampuan untuk membuat kluster EMR baru, untuk pengguna atau grup tersebut. Amazon EMR menyimpan pemetaan kebijakan sesi ini. Anda dapat memperbarui kebijakan sesi pengguna atau grup setelah penetapan.

> **Note**
>
> Izin terakhir untuk pengguna atau grup adalah persimpangan izin yang ditentukan dalam peran pengguna EMR Studio Anda dan izin yang ditentukan dalam kebijakan sesi untuk

pengguna atau grup tersebut. Jika pengguna termasuk dalam lebih dari satu grup yang ditetapkan ke Studio, EMR Studio menggunakan gabungan izin untuk pengguna tersebut.

Untuk menetapkan pengguna atau grup ke EMR Studio menggunakan konsol Amazon EMR

1. Arahkan ke konsol EMR Amazon baru dan pilih Beralih ke konsol lama dari navigasi samping. Untuk informasi selengkapnya tentang apa yang diharapkan saat beralih ke konsol lama, lihat Menggunakan konsol lama.
2. Pilih EMR Studio dari navigasi kiri.
3. Pilih nama Studio Anda dari daftar Studio, atau pilih Studio dan pilih Tampilkan detail, untuk membuka halaman detail Studio.
4. Pilih Tambahkan Pengguna untuk melihat tabel pencarian Pengguna dan Grup.
5. Pilih tab Pengguna atau Grup, dan masukkan istilah pencarian di bilah pencarian untuk menemukan pengguna atau grup.
6. Pilih satu atau beberapa pengguna atau grup dari daftar hasil pencarian. Anda dapat beralih bolak-balik antara tab Pengguna dan tab Grup.
7. Setelah Anda memilih pengguna dan grup untuk ditambahkan ke Studio, pilih Tambah. Anda akan melihat pengguna dan grup muncul di daftar Pengguna Studio. Mungkin diperlukan waktu beberapa detik agar daftar diperbarui.
8. Ikuti instruksi di Memperbarui izin untuk pengguna atau grup yang ditetapkan ke Studio untuk menyempurnakan izin Studio bagi pengguna atau grup.

Untuk menetapkan pengguna atau grup ke EMR Studio menggunakan AWS CLI

Masukkan nilai Anda sendiri untuk argumen `create-studio-session-mapping` berikut. Untuk informasi lebih lanjut tentang perintah `create-studio-session-mapping`, lihat Referensi Perintah AWS CLI.

- **`--studio-id`**— ID Studio yang ingin Anda tetapkan pengguna atau grup. Untuk petunjuk tentang cara mengambil ID Studio, lihatMelihat detail Studio.
- **`--identity-name`**— Nama pengguna atau grup dari Toko Identitas. Untuk informasi selengkapnya, lihat UserNameuntuk pengguna dan DisplayNamegrup di Referensi API Identity Store.
- **`--identity-type`** – Gunakan `USER` atau `GROUP` untuk menentukan jenis identitas.

- **`--session-policy-arn`** – Amazon Resource Name (ARN) untuk kebijakan sesi yang ingin Anda kaitkan dengan pengguna atau grup. Misalnya, **`arn:aws:iam::`*`<aws-account-id>`*`:policy/`*`EMRStudio_Advanced_User_Policy`**. Untuk informasi selengkapnya, lihat Membuat kebijakan izin untuk pengguna EMR Studio.

```
aws emr create-studio-session-mapping \
 --studio-id <example-studio-id> \
 --identity-name <example-identity-name> \
 --identity-type <USER-or-GROUP> \
 --session-policy-arn <example-session-policy-arn>
```

> **Note**
>
> Karakter lanjutan baris Linux (\) disertakan agar mudah dibaca. Karakter ini bisa dihapus atau digunakan dalam perintah Linux. Untuk Windows, hapus atau ganti dengan tanda sisipan (^).

Gunakan perintah `get-studio-session-mapping` untuk memverifikasi tugas baru. Ganti *< example-identity-name >* dengan nama Pusat Identitas IAM pengguna atau grup yang Anda perbarui.

```
aws emr get-studio-session-mapping \
 --studio-id <example-studio-id> \
 --identity-type <USER-or-GROUP> \
 --identity-name <user-or-group-name> \
```

Memperbarui izin untuk pengguna atau grup yang ditetapkan ke Studio

IAM

Untuk memperbarui izin pengguna atau grup saat Anda menggunakan mode autentikasi IAM, gunakan IAM untuk mengubah kebijakan izin IAM yang dilampirkan pada identitas IAM Anda (pengguna, grup, atau peran).

Untuk informasi selengkapnya, lihat Izin pengguna untuk mode otentikasi IAM.

IAM Identity Center

Untuk memperbarui izin EMR Studio untuk pengguna atau grup menggunakan konsol

1. Arahkan ke konsol EMR Amazon baru dan pilih Beralih ke konsol lama dari navigasi samping. Untuk informasi selengkapnya tentang apa yang diharapkan saat beralih ke konsol lama, lihat Menggunakan konsol lama.
2. Pilih EMR Studio dari navigasi kiri.
3. Pilih nama Studio Anda dari daftar Studio, atau pilih Studio dan pilih Tampilkan detail, untuk membuka halaman detail Studio.
4. Di daftar Pengguna Studio pada halaman detail Studio, cari pengguna atau grup yang ingin Anda perbarui. Anda dapat mencari berdasarkan nama atau jenis identitas.
5. Pilih pengguna atau grup yang ingin Anda perbarui dan pilih Tetapkan kebijakan untuk membuka kotak dialog Kebijakan sesi.
6. Pilih kebijakan yang akan diterapkan ke pengguna atau grup yang Anda pilih pada langkah 5, dan pilih Terapkan kebijakan. Daftar Pengguna Studio harus menampilkan nama kebijakan di kolom Kebijakan sesi untuk pengguna atau grup yang diperbarui.

Untuk memperbarui izin EMR Studio untuk pengguna atau grup menggunakan AWS CLI

Masukkan nilai Anda sendiri untuk argumen `update-studio-session-mappings` berikut. Untuk informasi lebih lanjut tentang perintah `update-studio-session-mappings`, lihat Referensi Perintah AWS CLI.

```
aws emr update-studio-session-mapping \
 --studio-id <example-studio-id> \
 --identity-name <name-of-user-or-group-to-update> \
 --session-policy-arn <new-session-policy-arn-to-apply> \
 --identity-type <USER-or-GROUP> \
```

Gunakan perintah `get-studio-session-mapping` untuk memverifikasi penetapan kebijakan sesi baru. Ganti *< example-identity-name >* dengan nama Pusat Identitas IAM pengguna atau grup yang Anda perbarui.

```
aws emr get-studio-session-mapping \
 --studio-id <example-studio-id> \
 --identity-type <USER-or-GROUP> \
```

```
  --identity-name <user-or-group-name> \
```

Menghapus pengguna atau grup dari Studio

IAM

Untuk menghapus pengguna atau grup dari EMR Studio saat Anda menggunakan mode autentikasi IAM, Anda harus mencabut akses pengguna ke Studio dengan mengonfigurasi ulang kebijakan izin IAM pengguna.

Dalam contoh kebijakan berikut, asumsikan bahwa Anda memiliki EMR Studio dengan pasangan nilai kunci tag. `Team = Quality Assurance` Menurut kebijakan, pengguna dapat mengakses Studios yang ditandai dengan `Team` kunci yang nilainya sama dengan salah satu `Data Analytics` atau`Quality Assurance`. Untuk menghapus pengguna dari Studio yang ditandai dengan`Team = Quality Assurance`, hapus `Quality Assurance` dari daftar nilai tag.

```
{
    "Version": "2012-10-17",
    "Statement": [
        {
            "Sid": "AllowCreateStudioPresignedUrl",
            "Effect": "Allow",
            "Action": [
                "elasticmapreduce:CreateStudioPresignedUrl"
            ],
            "Resource": "arn:aws:elasticmapreduce:<region>:<account-id>:studio/*",
            "Condition": {
                "StringEquals": {
                    "emr:ResourceTag/Team": [
                        "Data Analytics",
                        "Quality Assurance"
                  ]
                }
            }
        }
    ]
}
```

IAM Identity Center

Untuk menghapus pengguna atau grup dari EMR Studio menggunakan konsol

1. Arahkan ke konsol EMR Amazon baru dan pilih Beralih ke konsol lama dari navigasi samping. Untuk informasi selengkapnya tentang apa yang diharapkan saat beralih ke konsol lama, lihat Menggunakan konsol lama.
2. Pilih EMR Studio dari navigasi kiri.
3. Pilih nama Studio Anda dari daftar Studio, atau pilih Studio dan pilih Tampilkan detail, untuk membuka halaman detail Studio.
4. Di daftar Pengguna Studio pada halaman detail Studio, temukan pengguna atau grup yang ingin Anda hapus dari Studio. Anda dapat mencari berdasarkan nama atau jenis identitas.
5. Pilih pengguna atau grup yang ingin Anda hapus, pilih Hapus dan konfirmasi. Pengguna atau grup yang dihapus menghilang dari daftra Pengguna Studio.

Untuk menghapus pengguna atau grup dari EMR Studio menggunakan AWS CLI

Masukkan nilai Anda sendiri untuk argumen `delete-studio-session-mapping` berikut. Untuk informasi lebih lanjut tentang perintah `delete-studio-session-mapping`, lihat Referensi Perintah AWS CLI.

```
aws emr delete-studio-session-mapping \
 --studio-id <example-studio-id> \
 --identity-type <USER-or-GROUP> \
 --identity-name <name-of-user-or-group-to-delete> \
```

Mengelola Amazon EMR Studio

Bagian ini mencakup petunjuk untuk membantu Anda memantau, memperbarui, atau menghapus sumber daya EMR Studio. Untuk informasi tentang menetapkan pengguna atau memperbarui izin pengguna, lihat. Menetapkan dan mengelola pengguna EMR Studio

Melihat detail Studio

New console

Untuk melihat detail tentang EMR Studio dengan konsol baru

1. Buka konsol EMR Amazon di https://console.aws.amazon.com/emr.
2. Di bawah EMR Studio di navigasi kiri, pilih Studios.
3. Pilih Studio dari daftar Studios untuk membuka halaman detail Studio. Halaman detail Studio mencakup informasi Pengaturan Studio, seperti Deskripsi, VPC, dan Subnet Studio.

Old console

Untuk melihat detail tentang EMR Studio dengan konsol lama

1. Buka konsol EMR Amazon di https://console.aws.amazon.com/elasticmapreduce/home.
2. Pilih EMR Studio dari navigasi kiri.
3. Pilih Studio dari daftar Studios untuk membuka halaman detail Studio. Halaman detail Studio mencakup informasi Pengaturan Studio, seperti Deskripsi, VPC, dan Subnet Studio.

CLI

Untuk mengambil detail untuk EMR Studio by Studio ID menggunakan AWS CLI

Gunakan perintah `describe-studio` AWS CLI berikut untuk mengambil informasi mendetail tentang EMR Studio tertentu. Untuk informasi selengkapnya, lihat Referensi Perintah AWS CLI.

```
aws emr describe-studio \
 --studio-id <id-of-studio-to-describe> \
```

Untuk mengambil daftar EMR Studios menggunakan AWS CLI

Gunakan perintah perintah `list-studios` AWS CLI berikut ini. Untuk informasi selengkapnya, lihat Referensi Perintah AWS CLI.

```
aws emr list-studios
```

Berikut ini adalah contoh nilai yang dikembalikan untuk perintah `list-studios` dalam format JSON.

```
{
    "Studios": [
        {
            "AuthMode": "IAM",
            "VpcId": "vpc-b21XXXXX",
            "Name": "example-studio-name",
            "Url": "https://es-7HWP74SNGDXXXXXXXXXXXXXX.emrstudio-prod.us-
east-1.amazonaws.com",
            "CreationTime": 1605672582.781,
            "StudioId": "es-7HWP74SNGDXXXXXXXXXXXXXX",
            "Description": "example studio description"
        }
    ]
}
```

Memantau tindakan Amazon EMR Studio

Lihat aktivitas EMR Studio dan API

EMR Studio terintegrasi denganAWS CloudTrail, layanan yang menyediakan catatan tindakan yang diambil oleh pengguna, oleh peran IAM, atau oleh layanan lain AWS di EMR Studio. CloudTrail menangkap panggilan API untuk EMR Studio sebagai acara. Anda dapat melihat acara menggunakan CloudTrail konsol di https://console.aws.amazon.com/cloudtrail/.

Peristiwa EMR Studio memberikan informasi seperti Studio atau pengguna IAM mana yang membuat permintaan, dan apa jenis permintaannya.

> **Note**
>
> Tindakan di klaster seperti menjalankan pekerjaan notebook tidak dipancarkanAWS CloudTrail.

Anda juga dapat membuat jejak untuk pengiriman CloudTrail acara EMR Studio secara berkelanjutan ke bucket Amazon S3. Untuk informasi selengkapnya, silakan lihat Panduan Pengguna AWS CloudTrail.

Contoh CloudTrail Event: pengguna Memanggil DescribeStudio API

Berikut ini adalah contoh AWS CloudTrail peristiwa yang dibuat ketika pengguna,`admin`, memanggil DescribeStudioAPI. CloudTrail mencatat nama pengguna sebagai`admin`.

> **Note**
>
> Untuk melindungi detail Studio, acara EMR Studio API untuk DescribeStudio mengecualikan nilai untuk. `responseElements`


```
{
   "eventVersion":"1.08",
   "userIdentity":{
      "type":"IAMUser",
      "principalId":"AIDXXXXXXXXXXXXXXXXX",
      "arn":"arn:aws:iam::653XXXXXXXXX:user/admin",
      "accountId":"653XXXXXXXXX",
      "accessKeyId":"AKIAIOSFODNN7EXAMPLE",
      "userName":"admin"
   },
   "eventTime":"2021-01-07T19:13:58Z",
   "eventSource":"elasticmapreduce.amazonaws.com",
   "eventName":"DescribeStudio",
   "awsRegion":"us-east-1",
   "sourceIPAddress":"72.XX.XXX.XX",
   "userAgent":"aws-cli/1.18.188 Python/3.8.5 Darwin/18.7.0 botocore/1.19.28",
   "requestParameters":{
      "studioId":"es-9O5XXXXXXXXXXXXXXXXXXXX"
   },
   "responseElements":null,
   "requestID":"0fxxxxxx-xxxx-xxxx-xxxx-xxxxxxxxxxxx",
   "eventID":"b0xxxxxx-xxxx-xxxx-xxxx-xxxxxxxxxxxx",
   "readOnly":true,
   "eventType":"AwsApiCall",
   "managementEvent":true,
   "eventCategory":"Management",
   "recipientAccountId":"653XXXXXXXXX"
}
```

Melihat aktivitas pengguna dan pekerjaan Spark

Untuk melihat aktivitas pekerjaan Spark oleh pengguna Amazon EMR Studio, Anda dapat mengonfigurasi peniruan pengguna pada klaster. Dengan peniruan pengguna, setiap pekerjaan Spark yang dikirimkan dari Workspace dikaitkan dengan pengguna Studio yang menjalankan kode.

Saat peniruan identitas pengguna diaktifkan, Amazon EMR membuat direktori pengguna HDFS di node utama klaster untuk setiap pengguna yang menjalankan kode di Workspace. Misalnya, jika pengguna `studio-user-1@example.com` menjalankan kode, Anda dapat terhubung ke node utama dan melihat bahwa `hadoop fs -ls /user` memiliki direktori untuk`studio-user-1@example.com`.

Untuk menyiapkan peniruan pengguna Spark, atur properti berikut dalam klasifikasi konfigurasi berikut:

- `core-site`
- `livy-conf`

```
[
    {
        "Classification": "core-site",
        "Properties": {
          "hadoop.proxyuser.livy.groups": "*",
          "hadoop.proxyuser.livy.hosts": "*"
        }
    },
    {
        "Classification": "livy-conf",
        "Properties": {
          "livy.impersonation.enabled": "true"
        }
    }
]
```

Untuk melihat halaman server riwayat, lihat Debug aplikasi dan pekerjaan dengan EMR Studio. Anda juga dapat terhubung ke node utama cluster menggunakan SSH untuk melihat antarmuka web aplikasi. Untuk informasi selengkapnya, lihat Melihat antarmuka web yang di-host pada klaster Amazon EMR.

Memperbarui Amazon EMR Studio

Setelah membuat EMR Studio, Anda dapat memperbarui atribut berikut menggunakan AWS CLI:

- Nama
- Deskripsi
- Lokasi S3 default
- Subnet

Untuk memperbarui Studio EMR menggunakan AWS CLI

Gunakan `update-studio` AWS CLI perintah untuk memperbarui EMR Studio. Untuk informasi selengkapnya, lihat Referensi Perintah AWS CLI.

> **Note**
>
> Anda dapat mengaitkan Studio dengan maksimal 5 subnet. Subnet ini harus milik VPC yang sama dengan Studio. Daftar ID subnet yang Anda kirimkan ke `update-studio` perintah dapat menyertakan ID subnet baru, tetapi juga harus menyertakan semua ID subnet yang sudah Anda kaitkan dengan Studio. Anda tidak dapat menghapus subnet dari Studio.

```
aws emr update-studio \
 --studio-id <example-studio-id-to-update> \
 --name <example-new-studio-name> \
 --subnet-ids <old-subnet-id-1 old-subnet-id-2 old-subnet-id-3 new-subnet-id> \
```

Untuk memverifikasi perubahan, gunakan perintah `describe-studio` AWS CLI dan tentukan ID Studio Anda. Untuk informasi selengkapnya, lihat Referensi Perintah AWS CLI.

```
aws emr describe-studio \
 --studio-id <id-of-updated-studio> \
```

Menghapus Amazon EMR Studio dan Ruang Kerja

Saat Anda menghapus Studio, EMR Studio menghapus semua penugasan pengguna dan grup Pusat Identitas IAM yang terkait dengan Studio.

> **Note**
>
> Saat Anda menghapus Studio, Amazon EMR tidak menghapus Ruang Kerja yang terkait dengan Studio tersebut. Anda harus menghapus Workspaces di Studio Anda secara terpisah.

Hapus Ruang Kerja

Console

Karena setiap EMR Studio Workspace adalah instance notebook EMR, Anda dapat menggunakan konsol manajemen EMR Amazon untuk menghapus Workspaces. Anda dapat menghapus Workspaces menggunakan konsol Amazon EMR sebelum atau setelah menghapus Studio

Untuk menghapus Workspace menggunakan konsol Amazon EMR

1. Arahkan ke konsol EMR Amazon baru dan pilih Beralih ke konsol lama dari navigasi samping. Untuk informasi selengkapnya tentang apa yang diharapkan saat beralih ke konsol lama, lihat Menggunakan konsol lama.
2. Pilih Notebook.
3. Pilih Workspace yang ingin Anda hapus.
4. Pilih Hapus, lalu pilih Hapus lagi untuk mengonfirmasi.
5. Ikuti petunjuk untuk Menghapus objek di Panduan Pengguna Amazon Simple Storage Service Console untuk menghapus file notebook yang terkait dengan Workspace yang dihapus dari Amazon S3.

EMR Studio UI

From the Workspace UI

Menghapus Workspace dan file cadangan terkait dari EMR Studio

1. Login ke EMR Studio dengan URL akses Studio dan pilih Worksapcedari navigasi kiri.
2. Temukan Workspace Anda dalam daftar, lalu pilih kotak centang di samping namanya. Anda dapat memilih beberapa Workspace untuk dihapus pada saat yang sama.
3. Pilih Hapus di kanan atas daftar Workspace dan konfirmasi bahwa Anda ingin menghapus Workspace yang dipilih. Pilih Hapus untuk mengonfirmasi.

4. Jika Anda ingin menghapus file notebook yang dikaitkan dengan Workspace yang dihapus dari Amazon S3, ikuti petunjuk untuk Menghapus objek di Panduan Pengguna Amazon Simple Storage Service Console. Jika Anda tidak membuat Studio, konsultasikan administrator Studio Anda untuk menentukan lokasi cadangan Amazon S3 untuk Workspace yang dihapus.

From the Workspaces list

Menghapus Workspace dan file cadangan terkait dari daftar Workspaces

1. Arahkan ke daftar Workspace di konsol.
2. Pilih Workspace yang ingin Anda hapus dari daftar dan kemudian pilih Tindakan.
3. Pilih Hapus.
4. Jika Anda ingin menghapus file notebook yang dikaitkan dengan Workspace yang dihapus dari Amazon S3, ikuti petunjuk untuk Menghapus objek di Panduan Pengguna Amazon Simple Storage Service Console. Jika Anda tidak membuat Studio, konsultasikan administrator Studio Anda untuk menentukan lokasi cadangan Amazon S3 untuk Workspace yang dihapus.

Menghapus EMR Studio

New console

Untuk menghapus EMR Studio dengan konsol baru

1. Buka konsol EMR Amazon di https://console.aws.amazon.com/emr.
2. Di bawah EMR Studio di navigasi kiri, pilih Studios.
3. Pilih Studio dari daftar Studios dengan sakelar di sebelah kiri nama Studio. Pilih Hapus.

Old console

Untuk menghapus EMR Studio dengan konsol lama

1. Buka konsol EMR Amazon di https://console.aws.amazon.com/elasticmapreduce/home.
2. Pilih EMR Studio dari navigasi kiri.
3. Pilih Studio dari daftar Studios dan pilih Hapus.

CLI

Untuk menghapus EMR Studio dengan AWS CLI

Gunakan `delete-studio` AWS CLI perintah untuk menghapus EMR Studio. Untuk informasi selengkapnya, lihat Referensi Perintah AWS CLI.

```
aws emr delete-studio --studio-id <id-of-studio-to-delete>
```

Menentukan grup keamanan untuk mengontrol lalu lintas jaringan EMR Studio

Tentang grup keamanan EMR Studio

Amazon EMR Studio menggunakan dua grup keamanan untuk mengontrol lalu lintas jaringan antara Workspace di Studio dan klaster Amazon EMR terlampir yang berjalan pada Amazon EC2:

- Grup keamanan mesin yang menggunakan port 18888 untuk berkomunikasi dengan cluster EMR Amazon terpasang yang berjalan di Amazon EC2.
- Grup keamanan Workspace yang terkait dengan Workspace di Studio. Grup keamanan ini mencakup aturan HTTPS keluar untuk memungkinkan Workspace merutekan lalu lintas ke internet dan harus mengizinkan lalu lintas keluar ke internet pada port 443 untuk mengaktifkan penautan repositori Git ke Workspace.

EMR Studio menggunakan grup keamanan ini selain grup keamanan yang terkait dengan klaster EMR yang terlampir ke Workspace.

Anda harus membuat grup keamanan ini ketika Anda menggunakan AWS CLI untuk membuat Studio.

> **Note**
>
> Anda dapat menyesuaikan grup keamanan untuk EMR Studio dengan aturan yang disesuaikan dengan lingkungan Anda, tetapi Anda harus menyertakan aturan yang tercantum di halaman ini. Grup keamanan Workspace Anda tidak dapat mengizinkan lalu lintas masuk, dan grup keamanan mesin harus mengizinkan lalu lintas masuk dari grup keamanan Workspace.

Menggunakan Grup Keamanan EMR Studio Default

Saat Anda menggunakan konsol EMR Amazon, Anda dapat memilih grup keamanan default berikut. Grup keamanan default dibuat oleh EMR Studio atas nama Anda, dan menyertakan aturan masuk dan keluar minimum yang diperlukan untuk Ruang Kerja di EMR Studio.

- `DefaultEngineSecurityGroup`
- `DefaultWorkspaceSecurityGroupGit` atau `DefaultWorkspaceSecurityGroupWithoutGit`

Prasyarat

Untuk membuat grup keamanan untuk EMR Studio, Anda memerlukan Amazon Virtual Private Cloud (VPC) untuk Studio. Anda memilih VPC ini saat membuat grup keamanan. Ini harus menjadi VPC yang sama yang Anda tentukan saat Anda membuat Studio. Jika Anda berencana untuk menggunakan Amazon Amazon EMR di EKS dengan EMR Studio, pilih VPC untuk node pekerja klaster Amazon EKS Anda.

Petunjuk

Ikuti petunjuk dalam Membuat grup keamanan di Panduan Pengguna Amazon EC2 untuk Instans Linux untuk membuat grup keamanan engine dan grup keamanan Workspace di VPC Anda. Grup keamanan harus menyertakan aturan yang dirangkum dalam tabel berikut.

Saat Anda membuat grup keamanan untuk EMR Studio, perhatikan ID untuk keduanya. Anda menentukan setiap grup keamanan menurut ID saat membuat Studio.

Grup keamanan mesin

EMR Studio menggunakan port 18888 untuk berkomunikasi dengan klaster terlampir.

Aturan masuk

Tipe	Protokol	Port	Tujuan	Deskripsi
TCP	TCP	18888	Grup keamanan Workspace EMR Studio Anda.	Memungkinkan lalu lintas dari sumber daya apa pun di grup keamanan Workspace untuk EMR Studio.

Grup keamanan Workspace

Grup keamanan ini terkait dengan Workspace di EMR Studio.

Aturan keluar

Tipe	Protokol	Port	Tujuan	Deskripsi
TCP	TCP	18888	Grup keamanan mesin EMR Studio Anda.	Memungkinkan lalu lintas ke sumber daya apa pun di grup keamanan Mesin untuk EMR Studio.
HTTPS	TCP	443	0.0.0.0/0	Izinkan lalu lintas ke internet untuk menautkan repositori Git yang dihosting publik ke Ruang Kerja.

Buat AWS CloudFormation template untuk Amazon EMR Studio

Tentang templat klaster EMR Studio

Anda dapat membuat AWS CloudFormation template untuk membantu pengguna EMR Studio meluncurkan kluster EMR Amazon baru di Workspace. CloudFormation template adalah file teks yang diformat dalam JSON atau YAMM. Dalam template, Anda menjelaskan setumpuk sumber AWS daya dan memberi tahu CloudFormation cara menyediakan sumber daya tersebut untuk Anda. Untuk EMR Studio, Anda dapat membuat satu atau beberapa templat yang menjelaskan klaster EMR Amazon.

Anda mengatur template Anda diAWS Service Catalog. AWS Service Catalogmemungkinkan Anda membuat dan mengelola layanan TI yang umum digunakan yang disebut produk diAWS. Anda mengumpulkan template Anda sebagai produk dalam portofolio yang Anda bagikan dengan pengguna EMR Studio Anda. Setelah Anda membuat template cluster, pengguna Studio dapat meluncurkan klaster baru untuk Workspace dengan salah satu template Anda. Pengguna harus memiliki izin untuk membuat cluster baru dari template. Anda dapat menyetel izin pengguna dalam kebijakan izin EMR Studio.

Untuk mempelajari lebih lanjut tentang CloudFormation templat, lihat Templat di Panduan AWS CloudFormation Pengguna. Untuk informasi selengkapnya tentang AWS Service Catalog, lihat Apa itu AWS Service Catalog.

Video berikut menunjukkan cara mengatur template cluster AWS Service Catalog untuk EMR Studio. Anda juga dapat mempelajari lebih lanjut di lingkungan Build a self-service untuk setiap lini bisnis menggunakan Amazon EMR dan Service Catalog posting blog.

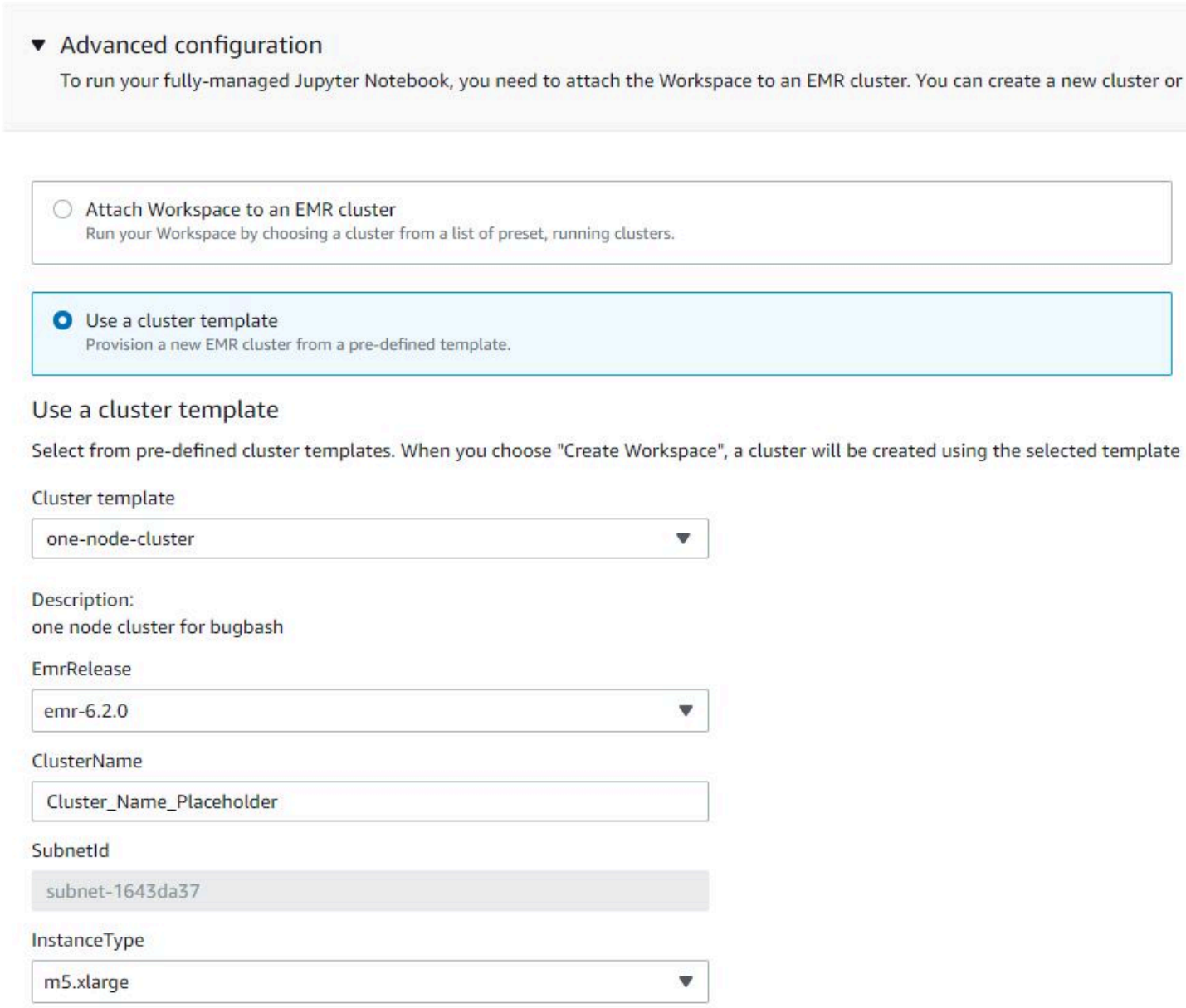
Parameter template opsional

Anda dapat menyertakan opsi tambahan di `Parameters`bagian template Anda. Parameter memungkinkan pengguna Studio memasukkan atau memilih nilai kustom untuk klaster. Misalnya, Anda dapat menambahkan parameter yang memungkinkan pengguna memilih rilis EMR Amazon tertentu. Untuk informasi selengkapnya, lihat Parameter dalam Panduan Pengguna AWS CloudFormation.

`Parameters`Bagian contoh berikut mendefinisikan parameter input tambahan seperti`ClusterName`, `EmrRelease` versi, dan`ClusterInstanceType`.

```
Parameters:
  ClusterName:
    Type: "String"
    Default: "Cluster_Name_Placeholder"
  EmrRelease:
    Type: "String"
    Default: "emr-6.2.0"
    AllowedValues:
    - "emr-6.2.0"
    - "emr-5.32.0"
  ClusterInstanceType:
    Type: "String"
    Default: "m5.xlarge"
    AllowedValues:
    - "m5.xlarge"
    - "m5.2xlarge"
```

Saat Anda menambahkan parameter, pengguna Studio melihat opsi formulir tambahan setelah memilih templat klaster. Gambar berikut menunjukkan opsi formulir tambahan untuk EmrReleaseversi, ClusterName, dan InstanceType.



Prasyarat

Sebelum Anda membuat template cluster, pastikan Anda memiliki izin IAM untuk mengakses tampilan konsol administrator Service Catalog. Anda juga memerlukan izin IAM yang diperlukan untuk melakukan tugas administratif Service Catalog. Untuk informasi selengkapnya, lihat Memberikan izin kepada administrator Service Catalog.

Petunjuk

Untuk membuat template cluster EMR menggunakan Service Catalog

1. Buat satu atau lebih CloudFormation template. Di mana Anda menyimpan template Anda terserah Anda. Karena template adalah file teks yang diformat, Anda dapat mengunggahnya ke Amazon S3 atau menyimpannya di sistem file lokal Anda. Untuk mempelajari lebih lanjut tentang CloudFormation templat, lihat Templat di Panduan AWS CloudFormation Pengguna.

   Gunakan aturan berikut untuk memberi nama templat Anda, atau memeriksa nama Anda terhadap pola `[a-zA-Z0-9][a-zA-Z0-9._-]*`.

   - Nama templat harus dimulai dengan huruf atau angka.
   - Nama templat hanya dapat terdiri dari huruf, angka, titik (.), garis bawah (_), dan tanda hubung (-).

   Setiap template cluster yang Anda buat harus menyertakan opsi berikut:

   Parameter masukan

   - ClusterName — Nama untuk cluster untuk membantu pengguna mengidentifikasinya setelah disediakan.

   Keluaran

   - `ClusterId`— ID dari cluster EMR yang baru disediakan.

   Berikut ini adalah contoh AWS CloudFormation template dalam format YAMAL untuk cluster dengan dua node. Contoh template mencakup opsi template yang diperlukan dan mendefinisikan parameter input tambahan untuk `EmrRelease` dan`ClusterInstanceType`.

```
awsTemplateFormatVersion: 2010-09-09

Parameters:
  ClusterName:
    Type: "String"
    Default: "Example_Two_Node_Cluster"
  EmrRelease:
    Type: "String"
```

```
    Default: "emr-6.2.0"
    AllowedValues:
    - "emr-6.2.0"
    - "emr-5.32.0"
  ClusterInstanceType:
    Type: "String"
    Default: "m5.xlarge"
    AllowedValues:
    - "m5.xlarge"
    - "m5.2xlarge"

Resources:
  EmrCluster:
    Type: AWS::EMR::Cluster
    Properties:
      Applications:
      - Name: Spark
      - Name: Livy
      - Name: JupyterEnterpriseGateway
      - Name: Hive
      EbsRootVolumeSize: '10'
      Name: !Ref ClusterName
      JobFlowRole: EMR_EC2_DefaultRole
      ServiceRole: EMR_DefaultRole_V2
      ReleaseLabel: !Ref EmrRelease
      VisibleToAllUsers: true
      LogUri:
        Fn::Sub: 's3://aws-logs-${AWS::AccountId}-${AWS::Region}/elasticmapreduce/'
      Instances:
        TerminationProtected: false
        Ec2SubnetId: 'subnet-ab12345c'
        MasterInstanceGroup:
          InstanceCount: 1
          InstanceType: !Ref ClusterInstanceType
        CoreInstanceGroup:
          InstanceCount: 1
          InstanceType: !Ref ClusterInstanceType
          Market: ON_DEMAND
          Name: Core

Outputs:
  ClusterId:
    Value:
      Ref: EmrCluster
```

```
    Description: The ID of the  EMR cluster
```

2. Buat portofolio untuk templat klaster Anda di akun AWS yang sama dengan Studio Anda.

   a. Buka konsol AWS Service Catalog di https://console.aws.amazon.com/servicecatalog/.

   b. Pilih Portofolio di menu navigasi kiri.

   c. Masukkan informasi yang diminta di halaman Buat portofolio.

   d. Pilih Buat. AWS Service Catalog membuat portofolio dan menampilkan detail portofolio.

3. Gunakan langkah-langkah berikut untuk menambahkan template cluster Anda sebagai AWS Service Catalog produk.

   a. Arahkan ke halaman Produk di bawah Administrasi di konsol AWS Service Catalog manajemen.

   b. Pilih Unggah produk baru.

   c. Masukkan nama produk dan pemilik.

   d. Tentukan file template Anda di bawah Detail versi.

   e. Pilih Tinjau untuk meninjau setelan produk Anda, lalu pilih Buat produk.

4. Lengkapi langkah-langkah berikut untuk menambahkan produk Anda ke portofolio Anda.

   a. Arahkan ke halaman Produk di konsol AWS Service Catalog manajemen.

   b. Pilih produk Anda, pilih Tindakan, lalu pilih Tambahkan produk ke portofolio.

   c. Pilih portofolio Anda, lalu pilih Tambahkan produk ke portofolio.

5. Buat kendala peluncuran untuk produk Anda. Batasan peluncuran adalah peran IAM yang menentukan izin pengguna untuk meluncurkan produk. Anda dapat menyesuaikan batasan peluncuran, tetapi harus mengizinkan izin untuk digunakan, CloudFormation Amazon EMR, dan. AWS Service Catalog Untuk informasi dan petunjuk selengkapnya, lihat kendala peluncuran Service Catalog.

6. Terapkan batasan peluncuran Anda ke setiap produk dalam portofolio Anda. Anda harus menerapkan batasan peluncuran untuk setiap produk secara individual.

   a. Pilih portofolio Anda dari halaman Portofolio di konsol AWS Service Catalog manajemen.

   b. Pilih tab Batasan dan pilih Buat batasan.

   c. Pilih produk Anda dan pilih Launch di bawah Constraint type. Pilih Lanjutkan.

   d. Pilih peran kendala peluncuran Anda di bagian Batasan peluncuran, lalu pilih Buat.

7. Berikan akses ke portofolio Anda.

a. Pilih portofolio Anda dari halaman Portofolio di konsol AWS Service Catalog manajemen.
b. Buka tab Grup, peran, dan pengguna dan pilih Tambahkan grup, peran, pengguna.
c. Cari peran IAM EMR Studio Anda di tab Peran, pilih peran Anda, dan pilih Tambahkan akses.

Jika Anda menggunakan...	Berikan akses ke...
Autentikasi IAM	Pengguna asli Anda
Federasi IAM	Peran IAM Anda untuk federasi
Federasi Pusat Identitas IAM	Peran pengguna EMR Studio Anda

Membuat akses dan izin untuk repositori berbasis Git

EMR Studio mendukung layanan berbasis Git berikut:

- AWS CodeCommit
- GitHub
- Bitbucket
- GitLab

Untuk memungkinkan pengguna EMR Studio mengaitkan repositori Git dengan Workspace, siapkan persyaratan akses dan izin berikut. Anda juga dapat mengonfigurasi repositori berbasis Git yang Anda host di jaringan privat dengan mengikuti petunjuk di Konfigurasikan repositori Git yang dihosting secara pribadi untuk EMR Studio.

Akses internet klaster

Kedua klaster Amazon EMR yang berjalan pada Amazon EC2 dan Amazon EMR di klaster EKS yang terlampir pada Workspace Studio harus berupa subnet privat yang menggunakan gateway terjemahan alamat jaringan (NAT), atau mereka harus dapat mengakses internet melalui gateway privat virtual. Untuk informasi selengkapnya, lihat Opsi Amazon VPC.

Grup keamanan yang Anda gunakan dengan EMR Studio juga harus menyertakan aturan keluar yang memungkinkan Workspace merutekan lalu lintas ke internet dari klaster EMR terlampir.

Untuk informasi selengkapnya, lihat Menentukan grup keamanan untuk mengontrol lalu lintas jaringan EMR Studio.

> **Important**
>
> Jika antarmuka jaringan berada dalam subnet publik, itu tidak akan dapat berkomunikasi dengan internet melalui Internet Gateway (IGW).

Izin untuk AWS Secrets Manager

Untuk memungkinkan pengguna EMR Studio mengakses repositori Git dengan rahasia yang disimpanAWS Secrets Manager, tambahkan kebijakan izin ke peran layanan untuk EMR Studio yang memungkinkan operasi. `secretsmanager:GetSecretValue`

Untuk informasi tentang cara menautkan repositori berbasis Git ke Workspace, lihat Menautkan repositori berbasis Git ke Workspace EMR Studio.

Konfigurasikan repositori Git yang dihosting secara pribadi untuk EMR Studio

Gunakan petunjuk berikut untuk mengonfigurasi repositori yang dihosting secara pribadi untuk Amazon EMR Studio. Berikan file konfigurasi dengan informasi tentang server DNS dan Git Anda. EMR Studio menggunakan informasi ini untuk mengonfigurasi Workspace yang dapat merutekan lalu lintas ke repositori yang dikelola sendiri.

> **Note**
>
> Jika Anda mengonfigurasi`DnsServerIpV4`, EMR Studio menggunakan server DNS Anda untuk menyelesaikan titik akhir EMR Amazon `GitServerDnsName` Anda dan Anda, seperti. `elasticmapreduce.us-east-1.amazonaws.com` Untuk menyiapkan endpoint untuk Amazon EMR, sambungkan ke endpoint Anda melalui VPC yang Anda gunakan dengan Studio Anda. Ini memastikan bahwa titik akhir EMR Amazon menyelesaikan ke IP pribadi. Untuk informasi selengkapnya, lihat Connect ke Amazon EMR menggunakan VPC endpoint antar muka.

Prasyarat

Sebelum mengonfigurasi repositori Git yang dihosting secara pribadi untuk EMR Studio, Anda memerlukan lokasi penyimpanan Amazon S3 tempat EMR Studio dapat mencadangkan file Workspaces dan notebook di Studio. Gunakan bucket S3 yang sama dengan yang Anda tentukan saat Anda membuat Studio.

Untuk mengonfigurasi satu atau beberapa repositori Git yang dihosting secara pribadi untuk EMR Studio

1. Buat file konfigurasi menggunakan template berikut. Sertakan nilai berikut untuk setiap server Git yang ingin Anda tentukan dalam konfigurasi Anda:

   - **`DnsServerIpV4`** - Alamat IPv4 dari server DNS Anda. Jika Anda memberikan nilai untuk keduanya `DnsServerIpV4` dan`GitServerIpV4List`, nilai untuk `DnsServerIpV4` diutamakan dan EMR Studio gunakan `DnsServerIpV4` untuk menyelesaikannya. `GitServerDnsName`

     > **Note**
     >
     > Untuk menggunakan repositori Git yang dihosting secara pribadi, server DNS Anda harus mengizinkan akses masuk dari EMR Studio. Kami mendorong Anda untuk mengamankan server DNS Anda dari akses lain yang tidak sah.

   - **`GitServerDnsName`** - Nama DNS server Git Anda. Misalnya `"git.example.com"`.
   - **`GitServerIpV4List`** - Daftar alamat IPv4 milik server Git Anda.

```
[
    {
        "Type": "PrivatelyHostedGitConfig",
        "Value": [
            {
                "DnsServerIpV4": "<10.24.34.xxx>",
                "GitServerDnsName": "<enterprise.git.com>",
                "GitServerIpV4List": [
                    "<xxx.xxx.xxx.xxx>",
                    "<xxx.xxx.xxx.xxx>"
                ]
            },
            {
                "DnsServerIpV4": "<10.24.34.xxx>",
```

```
                "GitServerDnsName": "<git.example.com>",
                "GitServerIpV4List": [
                    "<xxx.xxx.xxx.xxx>",
                    "<xxx.xxx.xxx.xxx>"
                ]
            }
        ]
    }
]
```

2. Simpan file konfigurasi Anda sebagai `configuration.json`.
3. Unggah file konfigurasi ke lokasi penyimpanan Amazon S3 Anda dalam folder bernama. `life-cycle-configuration` Misalnya, jika lokasi S3 default Anda`s3://DOC-EXAMPLE-BUCKET/studios`, file konfigurasi Anda akan masuk`s3://DOC-EXAMPLE-BUCKET/studios/`**`life-cycle-configuration/configuration.json`**.

> **Important**
>
> Kami mendorong Anda untuk membatasi akses ke `life-cycle-configuration` folder Anda ke administrator Studio dan ke peran layanan EMR Studio Anda, dan bahwa Anda mengamankan `configuration.json` terhadap akses yang tidak sah. Untuk instruksi, lihat Mengontrol akses ke bucket dengan kebijakan pengguna atau Praktik Terbaik Keamanan untuk Amazon S3.

Untuk instruksi pengunggahan, lihat Membuat folder dan Pengunggahan objek dalam Panduan Pengguna Amazon Storage Service. Untuk menerapkan konfigurasi ke Workspace yang ada, tutup dan mulai ulang Workspace setelah Anda mengunggah file konfigurasi ke Amazon S3.

Optimalkan lowongan kerja Spark di EMR Studio

Saat menjalankan pekerjaan Spark menggunakan EMR Studio, ada beberapa langkah yang dapat Anda ambil untuk membantu memastikan bahwa Anda mengoptimalkan sumber daya kluster Amazon EMR Anda.

Perpanjang sesi Livy Anda

Jika Anda menggunakan Apache Livy bersama dengan Spark di cluster EMR Amazon Anda, kami sarankan Anda meningkatkan batas waktu sesi Livy Anda dengan melakukan salah satu hal berikut:

- Saat Anda membuat klaster EMR Amazon, atur klasifikasi konfigurasi ini di bidang Enter Configuration.

```
[
    {
        "Classification": "livy-conf",
        "Properties": {
          "livy.server.session.timeout": "8h"
        }
    }
]
```

- Untuk kluster EMR yang sudah berjalan, sambungkan ke klaster Anda menggunakan `ssh` dan atur klasifikasi konfigurasi. `livy-conf` `/etc/livy/conf/livy.conf`

```
[
    {
        "Classification": "livy-conf",
        "Properties": {
          "livy.server.session.timeout": "8h"
        }
    }
]
```

  Anda mungkin perlu me-restart Livy setelah mengubah konfigurasi.

- Jika Anda tidak ingin sesi Livy Anda habis sama sekali, atur properti `livy.server.session.timeout-check` ke `false` dalam`/etc/livy/conf/livy.conf`.

Jalankan Spark dalam mode cluster

Dalam mode cluster, driver Spark berjalan pada node inti bukan pada node utama, meningkatkan pemanfaatan sumber daya pada node utama.

Untuk menjalankan aplikasi Spark Anda dalam mode cluster alih-alih mode klien default, pilih mode Cluster saat Anda mengatur mode Deploy saat mengonfigurasi langkah Spark Anda di cluster EMR Amazon baru Anda. Untuk informasi lebih lanjut, lihat Ikhtisar mode dalam dokumentasi Apache Spark.

Meningkatkan memori driver Spark

Untuk meningkatkan memori driver Spark, konfigurasikan sesi Spark Anda menggunakan perintah `%%configure` ajaib di notebook EMR Anda, seperti pada contoh berikut.

```
%%configure -f
{"driverMemory": "6000M"}
```

Menggunakan Amazon EMR Studio

Bagian ini berisi topik yang membantu Anda mengonfigurasi dan berinteraksi dengan Amazon EMR Studio.

Video berikut mencakup informasi praktis seperti cara membuat Workspace baru, dan cara meluncurkan klaster EMR Amazon baru dengan template cluster. Video ini juga berjalan melalui contoh notebook.

Bagian ini mencakup topik-topik berikut untuk membantu Anda bekerja di EMR Studio:

- Pelajari dasar-dasar Ruang Kerja
- Konfigurasikan kolaborasi Workspace
- Jalankan EMR Studio Workspace dengan peran runtime
- Jalankan notebook Workspace secara terprogram
- Jelajahi data dengan SQL Explorer
- Lampirkan komputasi ke Ruang Kerja EMR Studio
- Menautkan repositori berbasis Git ke Workspace EMR Studio
- Gunakan editor SQL Amazon Athena di EMR Studio
- CodeWhisperer Integrasi Amazon dengan EMR Studio Workspaces
- Debug aplikasi dan pekerjaan dengan EMR Studio
- Instal kernel dan pustaka di Ruang Kerja EMR Studio
- Tingkatkan kernel dengan perintah magic
- Gunakan notebook multi-bahasa dengan kernel Spark

Pelajari dasar-dasar Ruang Kerja

Saat menggunakan EMR Studio, Anda dapat membuat dan mengonfigurasi Ruang Kerja yang berbeda untuk mengatur dan menjalankan buku catatan. Bagian ini mencakup pembuatan dan bekerja dengan Workspaces. Untuk ikhtisar konseptual, lihat Workspace di Cara Kerja Amazon EMR Studio halaman.

Bagian ini mencakup topik-topik berikut untuk membantu Anda menggunakan Workspace EMR Studio:

- Membuat Workspace EMR Studio
- Meluncurkan Workspace
- Memahami antarmuka pengguna Workspace
- Jelajahi contoh buku catatan
- Menyimpan konten Workspace
- Menghapus file Workspace dan notebook
- Memahami status Workspace
- Mengatasi masalah konektivitas Workspace

Membuat Workspace EMR Studio

Anda dapat membuat Workspace EMR Studio untuk menjalankan kode notebook menggunakan antarmuka EMR Studio.

Membuat Workspace di EMR Studio

1. Masuk ke EMR Studio Anda.
2. Pilih Buat Ruang Kerja.
3. Masukkan Nama Workspace dan Deskripsi. Penamaan Workspace membantu Anda mengidentifikasinya di halaman Workspaces.
4. Jika Anda ingin bekerja dengan pengguna Studio lain di Workspace ini secara real time, aktifkan kolaborasi Workspace. Anda dapat mengonfigurasi kolaborator setelah meluncurkan Workspace.
5. Jika Anda ingin melampirkan cluster ke Workspace, perluas bagian Konfigurasi lanjutan. Anda dapat melampirkan cluster nanti, jika Anda mau. Untuk informasi selengkapnya, lihat Lampirkan komputasi ke Ruang Kerja EMR Studio.

> **Note**
>
> Untuk menyediakan klaster baru, Anda memerlukan izin akses dari administrator Anda.

Pilih salah satu opsi cluster untuk Workspace dan lampirkan cluster. Untuk informasi lebih lanjut tentang penyediaan klaster ketika Anda membuat Workspace, lihat Membuat dan melampirkan cluster EMR baru ke EMR Studio Workspace.

6. Pilih Buat Ruang Kerja di kanan bawah halaman.

Setelah Anda membuat Workspace, EMR Studio akan membuka halaman Workspaces. Anda akan melihat spanduk sukses hijau di bagian atas halaman dan dapat menemukan Workspace yang baru dibuat dalam daftar.

Secara default, Workspace dibagikan dan dapat dilihat oleh semua pengguna Studio. Namun, hanya satu pengguna yang dapat membuka dan bekerja di Workspace pada satu waktu. Untuk bekerja secara bersamaan dengan pengguna lain, Anda bisa Konfigurasikan kolaborasi Workspace

Meluncurkan Workspace

Untuk mulai bekerja dengan file notebook, luncurkan Workspace untuk mengakses editor notebook. Halaman Ruang Kerja di Studio mencantumkan semua Ruang Kerja yang dapat Anda akses dengan detail termasuk Nama, Status, Waktu pembuatan, dan Terakhir dimodifikasi.

> **Note**
>
> Jika Anda memiliki notebook EMR di konsol EMR Amazon lama, Anda dapat menemukannya di konsol baru sebagai EMR Studio Workspaces. Pengguna EMR Notebooks memerlukan izin peran IAM tambahan untuk mengakses atau membuat Ruang Kerja. Jika Anda baru saja membuat buku catatan di konsol lama, Anda mungkin perlu menyegarkan daftar Workspaces untuk melihatnya di konsol baru. Untuk informasi selengkapnya tentang transisi, lihat Amazon EMR Notebooks tersedia sebagai Amazon EMR Studio Workspaces di konsol baru dan Apa yang baru dengan konsol?

Untuk meluncurkan Workspace untuk mengedit dan menjalankan notebook

1. Pada halaman Workspaces Studio Anda, temukan Workspace. Anda dapat memfilter daftar dengan kata kunci atau dengan nilai kolom.
2. Pilih nama Workspace untuk meluncurkan Workspace di tab browser baru. Mungkin perlu waktu beberapa menit agar Workspace terbuka jika dalam keadaan Diam. Atau, pilih baris untuk Workspace dan kemudian pilih Launch Workspace. Anda dapat memilih dari opsi peluncuran berikut:

   - Peluncuran cepat - Luncurkan Workspace Anda dengan cepat dengan opsi default. Pilih Peluncuran cepat jika Anda ingin melampirkan cluster ke Workspace di. JupyterLab
   - Luncurkan dengan opsi - Luncurkan Ruang Kerja Anda dengan opsi khusus. Anda dapat memilih untuk meluncurkan di Jupyter atau JupyterLab, melampirkan Workspace Anda ke klaster EMR, dan memilih grup keamanan Anda.

> **Note**
>
> Hanya satu pengguna dapat membuka dan bekerja di Workspace pada satu waktu. Jika Anda memilih Workspace yang sudah digunakan, EMR Studio akan menampilkan notifikasi saat Anda mencoba membukanya. Kolom Pengguna pada halaman Workspaces menunjukkan pengguna yang bekerja di Workspace.

Memahami antarmuka pengguna Workspace

Antarmuka pengguna EMR Studio Workspace didasarkan pada JupyterLabantarmuka dengan tab yang dilambangkan ikon di bilah sisi kiri. Saat Anda mengarahkan kursor di atas ikon, Anda dapat melihat tooltip yang menunjukkan nama tab. Pilih tab dari bar sisi kiri untuk mengakses panel berikut.

- File Browser - Menampilkan file dan direktori di Workspace, serta file dan direktori dari repositori Git tertaut.
- Menjalankan Kernel dan Terminal - Daftar semua kernel dan terminal yang berjalan di Workspace. Untuk informasi selengkapnya, lihat Mengelola kernel dan terminal dalam JupyterLab dokumentasi resmi.
- Git — Menyediakan antarmuka pengguna grafis untuk melakukan perintah di repositori Git yang dilampirkan ke Workspace. Panel ini adalah JupyterLab ekstensi yang disebut jupyterlab-git. Untuk informasi lebih lanjut, lihat jupyterlab-git.

- Kluster EMR — Memungkinkan Anda melampirkan cluster ke atau melepaskan cluster dari Workspace untuk menjalankan kode notebook. Panel konfigurasi cluster EMR juga menyediakan opsi konfigurasi lanjutan untuk membantu Anda membuat dan melampirkan cluster baru ke Workspace. Untuk informasi selengkapnya, lihat Membuat dan melampirkan cluster EMR baru ke EMR Studio Workspace.
- Amazon EMR Git Repository - Membantu Anda menautkan Workspace dengan hingga tiga repositori Git. Untuk detail dan instruksinya, lihat Menautkan repositori berbasis Git ke Workspace EMR Studio.
- Contoh Notebook - Menyediakan daftar contoh buku catatan yang dapat Anda simpan ke Ruang Kerja. Anda juga dapat mengakses contoh dengan memilih Contoh Notebook di halaman Peluncur Ruang Kerja.
- Perintah — Menawarkan cara berbasis keyboard untuk mencari dan menjalankan perintah. JupyterLab Untuk informasi selengkapnya, lihat halaman Command palette di JupyterLab dokumentasi.
- Alat Notebook – Memungkinkan Anda memilih dan mengatur pilihan, seperti jenis sel slide dan metadata. Opsi Notebook Tools muncul di sidebar kiri setelah Anda membuka file notebook.
- Buka Tab — Daftar dokumen dan aktivitas terbuka di area kerja utama sehingga Anda dapat melompat ke tab yang terbuka. Untuk informasi selengkapnya, lihat halaman mode Tab dan dokumen tunggal dalam dokumentasi. JupyterLab
- Kolaborasi - Memungkinkan Anda mengaktifkan atau menonaktifkan kolaborasi Workspace, dan mengelola kolaborator. Untuk melihat panel Kolaborasi, Anda harus memiliki izin yang diperlukan. Untuk informasi selengkapnya, lihat Menetapkan kepemilikan untuk kolaborasi Workspace.

Jelajahi contoh buku catatan

Setiap EMR Studio Workspace menyertakan serangkaian contoh notebook yang dapat Anda gunakan untuk menjelajahi fitur EMR Studio. Untuk mengedit atau menjalankan contoh notebook, Anda dapat menyimpannya ke Workspace.

Untuk menyimpan contoh notebook ke Workspace

1. Dari bar sisi kiri, pilih tab Contoh Notebook untuk membuka panel Contoh Notebook. Anda juga dapat mengakses contoh dengan memilih Contoh Notebook di halaman Peluncur Ruang Kerja.
2. Pilih contoh notebook untuk melihat pratinjau di area kerja utama. Contohnya berformat hanya-baca.

3. Untuk menyimpan contoh notebook ke Workspace, pilih Save to Workspace. EMR Studio menyimpan contoh di direktori beranda Anda. Setelah menyimpan contoh notebook ke Workspace, Anda dapat mengganti nama, mengedit, dan menjalankannya.

Untuk informasi selengkapnya tentang contoh notebook, lihat repositori contoh GitHub Notebook EMR Studio.

Menyimpan konten Workspace

Saat Anda bekerja di editor notebook Workspace, EMR Studio menyimpan konten sel notebook dan output untuk Anda di lokasi Amazon S3 yang terkait dengan Studio. Proses pencadangan ini mempertahankan pekerjaan antar sesi.

Anda juga dapat menyimpan buku catatan dengan menekan CTRL+S di tab notebook terbuka atau dengan menggunakan salah satu opsi simpan di bawah File.

Cara lain untuk mencadangkan file notebook di Workspace adalah dengan mengaitkan Workspace dengan repositori berbasis Git dan menyinkronkan perubahan Anda dengan repositori jarak jauh. Melakukannya juga memungkinkan Anda menyimpan dan berbagi buku catatan dengan anggota tim yang menggunakan Ruang Kerja atau Studio yang berbeda. Untuk instruksi, lihat Menautkan repositori berbasis Git ke Workspace EMR Studio.

Menghapus file Workspace dan notebook

Saat menghapus file notebook dari EMR Studio Workspace, Anda menghapus file dari browser File, dan EMR Studio menghapus salinan cadangannya di Amazon S3. Anda tidak perlu mengambil langkah lebih lanjut untuk menghindari biaya penyimpanan saat menghapus file dari Workspace.

Saat Anda menghapus seluruh Workspace, file dan folder notebook akan tetap berada di lokasi penyimpanan Amazon S3. File terus bertambah biaya penyimpanan. Untuk menghindari biaya penyimpanan, hapus semua file dan folder cadangan yang terkait dengan Workspace yang dihapus dari Amazon S3.

Untuk menghapus file notebook dari Workspace EMR Studio

1. Pilih panel File browser dari sidebar kiri di Workspace.
2. Pilih file atau folder yang ingin Anda hapus. Klik kanan pada pilihan, lalu pilih Hapus. File menghilang dari daftar. EMR Studio menghapus file atau folder dari Amazon S3 untuk Anda.

From the Workspace UI

Menghapus Workspace dan file cadangan terkait dari EMR Studio

1. Login ke EMR Studio dengan URL akses Studio dan pilih Worksapcedari navigasi kiri.
2. Temukan Workspace Anda dalam daftar, lalu pilih kotak centang di samping namanya. Anda dapat memilih beberapa Workspace untuk dihapus pada saat yang sama.
3. Pilih Hapus di kanan atas daftar Workspace dan konfirmasi bahwa Anda ingin menghapus Workspace yang dipilih. Pilih Hapus untuk mengonfirmasi.
4. Jika Anda ingin menghapus file notebook yang dikaitkan dengan Workspace yang dihapus dari Amazon S3, ikuti petunjuk untuk Menghapus objek di Panduan Pengguna Amazon Simple Storage Service Console. Jika Anda tidak membuat Studio, konsultasikan administrator Studio Anda untuk menentukan lokasi cadangan Amazon S3 untuk Workspace yang dihapus.

From the Workspaces list

Menghapus Workspace dan file cadangan terkait dari daftar Workspaces

1. Arahkan ke daftar Workspace di konsol.
2. Pilih Workspace yang ingin Anda hapus dari daftar dan kemudian pilih Tindakan.
3. Pilih Hapus.
4. Jika Anda ingin menghapus file notebook yang dikaitkan dengan Workspace yang dihapus dari Amazon S3, ikuti petunjuk untuk Menghapus objek di Panduan Pengguna Amazon Simple Storage Service Console. Jika Anda tidak membuat Studio, konsultasikan administrator Studio Anda untuk menentukan lokasi cadangan Amazon S3 untuk Workspace yang dihapus.

Memahami status Workspace

Setelah Anda membuat EMR Studio Workspace, EMR Studio akan muncul sebagai baris dalam daftar Workspaces di Studio Anda dengan nama, status, waktu pembuatan, dan stempel waktu terakhir yang dimodifikasi. Tabel berikut menjelaskan status Workspace.

Status	Deskripsi
Memulai	Workspace sedang dipersiapkan, tetapi belum siap digunakan. Anda tidak dapat membuka Workspace ketika statusnya Memulai.
Siap	Anda dapat membuka Workspace untuk menggunakan editor notebook, tetapi Anda harus melampirkan Workspace ke klaster EMR sebelum dapat menjalankan kode notebook.
Melampirkan	Workspace sedang dilampirkan ke klaster.
Terlampir	Workspace terlampir ke klaster EMR dan siap bagi Anda untuk menulis dan menjalankan kode notebook. Jika status Workspace tidak Terlampir, Anda harus melampirkannya ke suatu klaster sebelum Anda dapat menjalankan kode notebook.
Menganggur	Ruang kerja telah berhenti. Untuk mengaktif kan kembali Workspace yang diam, pilih dari daftar Workspace. Status berubah dari Diam ke Memulai ke Siap ketika Anda memilih Workspace.
Berhenti	Workspace dimatikan dan akan diatur ke Idle. Saat Anda menghentikan Workspace , Workspace akan menghentikan kernel notebook yang sesuai. EMR Studio menghenti kan notebook yang tidak aktif untuk waktu lama.
Menghapus	Ketika Anda menghapus Workspace, EMR Studio menandainya untuk penghapusan dan memulai proses penghapusan. Setelah proses penghapusan selesai, Workspace menghilang dari daftar. Saat Anda menghapus Workspace

Status	Deskripsi
	, file notebook akan tetap berada di lokasi penyimpanan Amazon S3.

Mengatasi masalah konektivitas Workspace

Untuk mengatasi masalah konektivitas Workspace, Anda dapat menghentikan dan memulai ulang Workspace. Saat Anda me-restart Workspace, EMR Studio meluncurkan Workspace di Availability Zone yang berbeda atau subnet lain yang terkait dengan Studio Anda.

Untuk menghentikan dan memulai ulang Ruang Kerja EMR Studio

1. Tutup Workspace di browser Anda.
2. Arahkan ke daftar Workspace di konsol.
3. Pilih Workspace Anda dari daftar dan pilih Tindakan.
4. Pilih Berhenti dan tunggu status Workspace berubah dari Berhenti ke Idle.
5. Pilih Tindakan lagi, lalu pilih Mulai untuk memulai ulang Workspace.
6. Tunggu status Workspace berubah dari Mulai ke Siap, lalu pilih nama Workspace untuk membukanya kembali di tab browser baru.

Konfigurasikan kolaborasi Workspace

Kolaborasi ruang kerja memungkinkan Anda menulis dan menjalankan kode buku catatan secara bersamaan dengan anggota tim Anda yang lain. Saat Anda bekerja di file buku catatan yang sama, Anda akan melihat perubahan saat kolaborator membuatnya. Anda dapat mengaktifkan kolaborasi saat membuat Workspace, atau mengaktifkan dan menonaktifkan kolaborasi di Workspace yang ada.

> **Note**
>
> Kolaborasi EMR Studio Workspace tidak didukung dengan aplikasi interaktif EMR Tanpa Server atau jika propagasi identitas tepercaya diaktifkan.

Prasyarat

Sebelum Anda mengonfigurasi kolaborasi untuk Workspace, pastikan Anda menyelesaikan tugas-tugas berikut:

- Pastikan admin EMR Studio Anda telah memberi Anda izin yang diperlukan. Misalnya, pernyataan berikut memungkinkan pengguna mengonfigurasi kolaborasi untuk Workspace apa pun dengan kunci tag `creatorUserId` yang nilainya cocok dengan ID pengguna (ditunjukkan oleh variabel kebijakan`aws:userId`).

```
{
    "Sid": "UserRolePermissionsForCollaboration",
    "Action": [
        "elasticmapreduce:UpdateEditor",
        "elasticmapreduce:PutWorkspaceAccess",
        "elasticmapreduce:DeleteWorkspaceAccess",
        "elasticmapreduce:ListWorkspaceAccessIdentities"
    ],
    "Resource": "*",
    "Effect": "Allow",
    "Condition": {
        "StringEquals": {
            "elasticmapreduce:ResourceTag/creatorUserId": "${aws:userid}"
        }
    }
}
```

- Pastikan bahwa peran layanan yang terkait dengan EMR Studio Anda memiliki izin yang diperlukan untuk mengaktifkan dan mengonfigurasi kolaborasi Workspace, seperti pada pernyataan contoh berikut.

```
{
    "Sid": "AllowWorkspaceCollaboration",
    "Effect": "Allow",
    "Action": [
        "iam:GetUser",
        "iam:GetRole",
        "iam:ListUsers",
        "iam:ListRoles",
        "sso:GetManagedApplicationInstance",
        "sso-directory:SearchUsers"
    ],
```

```
    "Resource": "*"
}
```

Untuk informasi selengkapnya, lihat Membuat peran layanan EMR Studio.

Untuk mengaktifkan kolaborasi Workspace dan menambahkan kolaborator

1. Di Workspace Anda, pilih ikon Kolaborasi dari layar Launcher atau bagian bawah panel kiri.

   > **Note**
   >
   > Anda tidak akan melihat panel Kolaborasi kecuali admin Studio Anda telah memberi Anda izin untuk mengonfigurasi kolaborasi untuk Workspace. Untuk informasi selengkapnya, lihat Menetapkan kepemilikan untuk kolaborasi Workspace.

2. Pastikan toggle Izinkan kolaborasi Workspace berada di posisi aktif. Bila Anda mengaktifkan kolaborasi, hanya Anda dan kolaborator yang Anda tambahkan yang dapat melihat Workspace dalam daftar di halaman Studio Workspaces.
3. Masukkan nama Kolaborator. Ruang kerja Anda dapat memiliki maksimal lima kolaborator termasuk Anda sendiri. Kolaborator dapat berupa pengguna mana pun yang memiliki akses ke EMR Studio Anda. Jika Anda tidak memasukkan kolaborator, Workspace adalah Workspace pribadi yang hanya dapat diakses oleh Anda.

   Tabel berikut menentukan nilai kolaborator yang berlaku untuk dimasukkan berdasarkan tipe identitas pemilik.

   > **Note**
   >
   > Pemilik hanya dapat mengundang kolaborator dengan tipe identitas yang sama. Misalnya, pengguna hanya dapat menambahkan pengguna lain, dan pengguna Pusat Identitas IAM hanya dapat menambahkan pengguna Pusat Identitas IAM lainnya.

Mode autentikasi	Nilai untuk dimasukkan untuk nama Kolaborator
Autentikasi IAM	nama pengguna. Ini adalah nama yang dilihat pengguna saat masuk ke fileAWS Management Console.
Federasi IAM	Nama peran IAM dan nama sesi opsional. Untuk menambahkan semua pengguna federasi yang mengambil peran IAM yang sama, tentukan nama peran IAM untuk federasi. Untuk menambahkan satu pengguna sebagai kolaborator, tentukan peran dan nama sesi. Sebagai contoh, `MyRoleName:MySessionName` .
SSO	Nama pengguna IAM Identity Center seperti `user@example.com`.

4. Pilih Tambahkan. Kolaborator sekarang dapat melihat Workspace di halaman EMR Studio Workspaces mereka, dan meluncurkan Workspace untuk menggunakannya secara real time bersama Anda.

> **Note**
>
> Jika Anda menonaktifkan kolaborasi Workspace, Workspace akan kembali ke status bersama dan dapat dilihat oleh semua pengguna Studio. Dalam status bersama, hanya satu pengguna Studio yang dapat membuka dan bekerja di Workspace sekaligus.

Jalankan EMR Studio Workspace dengan peran runtime

> **Note**
>
> Fungsionalitas peran runtime yang dijelaskan di halaman ini hanya berlaku untuk Amazon EMR yang berjalan di Amazon EC2, dan tidak mengacu pada fungsionalitas peran runtime di aplikasi interaktif EMR Tanpa Server. Untuk mempelajari selengkapnya tentang cara menggunakan peran runtime di EMR Tanpa Server, lihat Peran runtime Job di Panduan Pengguna Tanpa Server Amazon EMR.

Peran runtime adalah peran AWS Identity and Access Management (IAM) yang dapat Anda tentukan saat mengirimkan pekerjaan atau kueri ke klaster EMR Amazon. Pekerjaan atau kueri yang Anda kirimkan ke kluster EMR menggunakan peran runtime untuk mengakses AWS sumber daya, seperti objek di Amazon S3.

Saat melampirkan EMR Studio Workspace ke kluster EMR yang menggunakan Amazon EMR 6.11 atau yang lebih tinggi, Anda dapat memilih peran runtime untuk pekerjaan atau kueri yang Anda kirimkan untuk digunakan saat mengakses sumber daya. AWS Namun, jika klaster EMR tidak mendukung peran runtime, klaster EMR tidak akan mengambil peran saat mengakses sumber daya. AWS

Sebelum Anda dapat menggunakan peran runtime dengan Amazon EMR Studio Workspace, administrator harus mengonfigurasi izin pengguna agar pengguna Studio dapat memanggil `elasticmapreduce:GetClusterSessionCredentials` API pada peran runtime. Kemudian, luncurkan cluster baru dengan peran runtime yang dapat Anda gunakan dengan Amazon EMR Studio Workspace.

Di halaman ini

Konfigurasikan izin pengguna untuk peran runtime

Konfigurasikan izin pengguna sehingga pengguna Studio dapat memanggil `elasticmapreduce:GetClusterSessionCredentials` API pada peran runtime yang ingin digunakan pengguna. Anda juga harus mengkonfigurasi the section called “Izin pengguna studio (EC2, EKS)” sebelum pengguna dapat mulai menggunakan Studio.

> **Warning**
>
> Untuk memberikan izin ini, buat kondisi berdasarkan kunci `elasticmapreduce:ExecutionRoleArn` konteks saat Anda memberikan akses pemanggil untuk memanggil `GetClusterSessionCredentials` API. Contoh berikut menunjukkan bagaimana melakukannya.

```
{
      "Sid": "AllowSpecificExecRoleArn",
      "Effect": "Allow",
      "Action": [
          "elasticmapreduce:GetClusterSessionCredentials"
      ],
      "Resource": "*",
      "Condition": {
          "StringEquals": {
              "elasticmapreduce:ExecutionRoleArn": [
                  "arn:aws:iam::111122223333:role/test-emr-demo1",
                  "arn:aws:iam::111122223333:role/test-emr-demo2"
              ]
          }
      }
  }
```

Contoh berikut menunjukkan bagaimana mengizinkan prinsipal IAM untuk menggunakan peran IAM bernama `test-emr-demo3` sebagai peran runtime. Selain itu, pemegang polis hanya akan dapat mengakses klaster EMR Amazon dengan ID cluster. `j-123456789`

```
{
    "Sid":"AllowSpecificExecRoleArn",
    "Effect":"Allow",
    "Action":[
```

```
        "elasticmapreduce:GetClusterSessionCredentials"
    ],
    "Resource": [
          "arn:aws:elasticmapreduce:<region>:111122223333:cluster/j-123456789"
     ],
    "Condition":{
        "StringEquals":{
            "elasticmapreduce:ExecutionRoleArn":[
                "arn:aws:iam::111122223333:role/test-emr-demo3"
            ]
        }
    }
}
```

Contoh berikut memungkinkan prinsipal IAM menggunakan peran IAM apa pun dengan nama yang dimulai dengan string `test-emr-demo4` sebagai peran runtime. Selain itu, pemegang polis hanya akan dapat mengakses kluster EMR Amazon yang ditandai dengan pasangan nilai kunci. `tagKey:tagValue`

```
{
    "Sid":"AllowSpecificExecRoleArn",
    "Effect":"Allow",
    "Action":[
        "elasticmapreduce:GetClusterSessionCredentials"
    ],
    "Resource": "*",
    "Condition":{
        "StringEquals":{
             "elasticmapreduce:ResourceTag/tagKey": "tagValue"
        },
        "StringLike":{
            "elasticmapreduce:ExecutionRoleArn":[
                "arn:aws:iam::111122223333:role/test-emr-demo4*"
            ]
        }
    }
}
```

Luncurkan cluster baru dengan peran runtime

Setelah Anda memiliki izin yang diperlukan, luncurkan klaster baru dengan peran runtime yang dapat Anda gunakan dengan Amazon EMR Studio Workspace.

Jika Anda telah meluncurkan cluster baru dengan peran runtime, Anda dapat melompat ke the section called “Gunakan klaster dengan Workspace Anda” bagian tersebut.

1. Pertama, lengkapi prasyarat di bagian ini. Peran runtime untuk langkah-langkah EMR Amazon
2. Kemudian, luncurkan cluster dengan pengaturan berikut untuk menggunakan peran runtime dengan Amazon EMR Studio Workspaces. Untuk petunjuk tentang cara meluncurkan klaster Anda, lihatMenentukan konfigurasi keamanan untuk sebuah klaster.
   - Pilih label rilis emr-6.11.0 atau yang lebih baru.
   - Pilih Spark, Livy, dan Jupyter Enterprise Gateway sebagai aplikasi cluster Anda.
   - Gunakan konfigurasi keamanan yang Anda buat pada langkah sebelumnya.
   - Secara opsional, Anda dapat mengaktifkan Lake Formation untuk cluster EMR Anda. Untuk informasi selengkapnya, lihat Aktifkan Lake Formation dengan Amazon EMR.

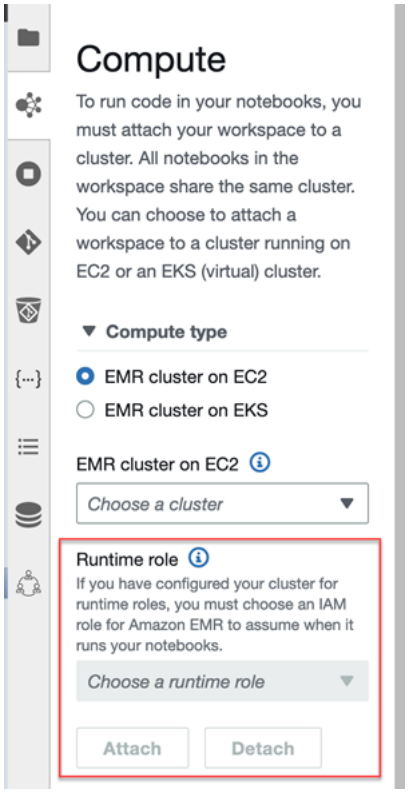
Setelah meluncurkan klaster, Anda siap menggunakan klaster berkemampuan peran runtime dengan EMR Studio Workspace.

> **Note**
>
> ExecutionRoleArnNilai saat ini tidak didukung dengan operasi StartNotebookExecutionAPI saat `ExecutionEngineConfig.Type` nilainya`EMR`.

Gunakan cluster EMR dengan peran runtime di Workspaces

Setelah menyiapkan dan meluncurkan klaster, Anda dapat menggunakan klaster berkemampuan peran runtime dengan EMR Studio Workspace.

1. Buat ruang kerja baru atau luncurkan ruang kerja yang ada. Untuk informasi selengkapnya, lihat Membuat Workspace EMR Studio.
2. Pilih tab klaster EMR di bilah sisi kiri Ruang Kerja terbuka Anda, perluas bagian Jenis komputasi, dan pilih klaster Anda dari cluster EMR di menu EC2, dan peran runtime dari menu peran Runtime.



3. Pilih Lampirkan untuk melampirkan cluster dengan peran runtime ke Workspace Anda.

Pertimbangan-pertimbangan

Perhatikan pertimbangan berikut saat Anda menggunakan klaster berkemampuan peran runtime dengan Amazon EMR Studio Workspace:

- Anda hanya dapat memilih peran runtime saat melampirkan EMR Studio Workspace ke kluster EMR yang menggunakan Amazon EMR rilis 6.11 atau yang lebih tinggi.
- Fungsionalitas peran runtime yang dijelaskan di halaman ini hanya didukung dengan Amazon EMR yang berjalan di Amazon EC2, dan tidak didukung dengan aplikasi interaktif EMR Tanpa Server. Untuk mempelajari lebih lanjut tentang peran runtime untuk EMR Tanpa Server, lihat Peran runtime Job di Panduan Pengguna Tanpa Server Amazon EMR.
- Meskipun Anda perlu mengonfigurasi izin tambahan sebelum dapat menentukan peran runtime saat mengirimkan pekerjaan ke klaster, Anda tidak memerlukan izin tambahan untuk mengakses file yang dihasilkan oleh EMR Studio Workspace. Izin untuk file tersebut sama dengan file yang dihasilkan dari cluster tanpa peran runtime.

- Anda tidak dapat menggunakan SQL Explorer di EMR Studio Workspace dengan cluster yang memiliki peran runtime. Amazon EMR menonaktifkan SQL Explorer di UI saat Workspace dilampirkan ke kluster EMR yang mendukung peran runtime.
- Anda tidak dapat menggunakan mode kolaborasi di EMR Studio Workspace dengan klaster yang memiliki peran runtime. Amazon EMR menonaktifkan kemampuan kolaborasi Workspace saat Workspace dilampirkan ke klaster EMR yang mendukung peran runtime. Workspace akan tetap dapat diakses hanya oleh pengguna yang melampirkan Workspace.
- Anda tidak dapat menggunakan peran runtime di Studio dengan propagasi identitas tepercaya IAM Identity Center diaktifkan.
- Anda mungkin menemukan peringatan “Halaman mungkin tidak aman!” dari Spark UI untuk cluster yang mendukung peran runtime. Jika ini terjadi, lewati peringatan untuk terus melihat UI Spark.

Jalankan notebook Workspace secara terprogram

> **Note**
>
> Eksekusi terprogram notebook tidak didukung dengan aplikasi interaktif Amazon EMR Serverless.

Anda dapat menjalankan notebook Amazon EMR Studio Workspace Anda secara terprogram dengan skrip atau di file. AWS CLI Untuk mempelajari cara menjalankan notebook Anda secara terprogram, lihat. Contoh perintah untuk menjalankan EMR Notebooks secara programatis

Jelajahi data dengan SQL Explorer

> **Note**
>
> SQL Explorer untuk EMR Studio tidak didukung dengan aplikasi interaktif Amazon EMR Tanpa Server atau di Studio dengan propagasi identitas tepercaya IAM Identity Center diaktifkan.

Topik ini memberikan informasi untuk membantu Anda memulai SQL Explorer di Amazon EMR Studio. SQL Explorer adalah alat satu halaman di Workspace Anda yang membantu Anda memahami sumber data dalam katalog data klaster EMR Anda. Anda dapat menggunakan

SQL Explorer untuk menelusuri data Anda, menjalankan kueri SQL untuk mengambil data, dan mengunduh hasil kueri.

SQL Explorer mendukung Presto. Sebelum Anda menggunakan SQL Explorer, pastikan Anda memiliki cluster yang menggunakan Amazon EMR versi 5.34.0 atau yang lebih baru atau versi 6.4.0 atau yang lebih baru dengan Presto diinstal. Amazon EMR Studio SQL Explorer tidak mendukung klaster Presto yang telah Anda konfigurasikan dengan enkripsi dalam perjalanan. Ini karena Presto berjalan dalam mode TLS pada cluster ini.

Jelajahi katalog data klaster Anda

SQL Explorer menyediakan antarmuka browser katalog yang dapat Anda gunakan untuk menjelajahi dan memahami bagaimana data Anda diatur. Misalnya, Anda dapat menggunakan browser katalog data untuk memverifikasi nama tabel dan kolom sebelum Anda menulis kueri SQL.

Untuk menelusuri katalog data Anda

1. Buka SQL Explorer di Workspace Anda.
2. Pastikan Workspace Anda terpasang ke klaster EMR yang berjalan di EC2 yang menggunakan Amazon EMR versi 6.4.0 atau yang lebih baru dengan Presto diinstal. Anda dapat memilih cluster yang ada, atau membuat yang baru. Untuk informasi selengkapnya, lihat Lampirkan komputasi ke Ruang Kerja EMR Studio.
3. Pilih Database dari daftar dropdown untuk dijelajahi.
4. Perluas tabel di database Anda untuk melihat nama kolom tabel. Anda juga dapat memasukkan kata kunci di bilah pencarian untuk memfilter hasil tabel.

Jalankan kueri SQL untuk mengambil data

Untuk mengambil data dengan query SQL dan men-download hasilnya

1. Buka SQL Explorer di Workspace Anda.
2. Pastikan Workspace Anda terpasang ke cluster EMR yang berjalan di EC2 dengan Presto dan Spark diinstal. Anda dapat memilih cluster yang ada, atau membuat yang baru. Untuk informasi selengkapnya, lihat Lampirkan komputasi ke Ruang Kerja EMR Studio.
3. Pilih Buka editor untuk membuka tab editor baru di Workspace Anda.
4. Tulis kueri SQL Anda di tab editor.
5. Pilih Jalankan.

6. Lihat hasil kueri Anda di bawah Pratinjau hasil. SQL Explorer menampilkan 100 hasil pertama secara default. Anda dapat memilih jumlah hasil yang berbeda untuk ditampilkan (hingga 1000) menggunakan menu menu pratinjau 100 hasil kueri pertama.
7. Pilih Unduh hasil untuk mengunduh hasil Anda dalam format CSV. Anda dapat mengunduh hingga 1000 baris hasil.

Lampirkan komputasi ke Ruang Kerja EMR Studio

Amazon EMR Studio menjalankan perintah notebook menggunakan kernel pada klaster EMR. Sebelum dapat memilih kernel, Anda harus melampirkan Workspace ke cluster yang menggunakan instans Amazon EC2, ke Amazon EMR di kluster EKS, atau ke aplikasi EMR Tanpa Server. EMR Studio memungkinkan Anda melampirkan Workspace ke klaster baru atau yang sudah ada, dan memberi Anda fleksibilitas untuk mengubah klster tanpa menutup Workspace.

Bagian ini membahas topik-topik berikut untuk membantu Anda menggunakan dan menyediakan klaster untuk EMR Studio:

- Melampirkan klaster Amazon EC2 ke Ruang Kerja EMR Studio
- Melampirkan EMR Amazon di kluster EKS ke EMR Studio Workspace
- Lampirkan aplikasi Amazon EMR Tanpa Server ke EMR Studio Workspace
- Membuat dan melampirkan cluster EMR baru ke EMR Studio Workspace
- Lepaskan komputasi dari EMR Studio Workspace

Melampirkan klaster Amazon EC2 ke Ruang Kerja EMR Studio

Anda dapat melampirkan kluster EMR yang berjalan di Amazon EC2 ke Workspace saat membuat Workspace, atau melampirkan cluster ke Workspace yang ada. Jika Anda ingin membuat dan melampirkan klaster baru, lihat Membuat dan melampirkan cluster EMR baru ke EMR Studio Workspace.

> **Note**
>
> Ruang kerja di Studio yang mengaktifkan propagasi identitas tepercaya IAM Identity Center hanya dapat dilampirkan ke klaster EMR dengan konfigurasi keamanan yang mengaktifkan Pusat Identitas.

On create

Lampirkan ke klaster komputasi Amazon EMR saat Anda membuat Workspace

1. Di kotak dialog Create a Workspace, pastikan Anda telah memilih subnet untuk Workspace baru. Perluas bagian Konfigurasi lanjutan.
2. Di kotak dialog Create a Workspace, pastikan Anda telah memilih subnet untuk Workspace baru. Perluas bagian Konfigurasi lanjutan.
3. Pilih Lampirkan Workspace ke klaster EMR.
4. Dalam daftar dropdown cluster EMR, pilih klaster EMR yang ada untuk dilampirkan ke Workspace.

Setelah Anda melampirkan cluster, selesaikan pembuatan Workspace. Saat Anda membuka Workspace baru untuk pertama kalinya dan memilih panel kluster EMR, Anda akan melihat cluster yang Anda pilih terpasang.

On launch

Lampirkan ke klaster komputasi EMR Amazon saat Anda meluncurkan Workspace

1. Arahkan ke daftar Workspaces dan pilih baris untuk Workspace yang ingin Anda luncurkan. Kemudian, pilih Luncurkan Ruang Kerja > Luncurkan dengan opsi.
2. Pilih klaster EMR untuk dilampirkan ke Workspace Anda.

Setelah Anda melampirkan cluster, selesaikan pembuatan Workspace. Saat Anda membuka Workspace baru untuk pertama kalinya dan memilih panel kluster EMR, Anda akan melihat cluster yang Anda pilih terpasang.

In JupyterLab

Melampirkan Workspace ke klaster komputasi Amazon EMR di JupyterLab

1. Pilih Workspace Anda, lalu pilih Launch Workspace > Quick launch.
2. Di dalam JupyterLab, buka tab Cluster di sidebar kiri.
3. Pilih EMR pada dropdown cluster EC2, atau pilih Amazon EMR di kluster EKS.
4. Pilih Lampirkan untuk melampirkan cluster ke Workspace Anda.

Setelah Anda melampirkan cluster, selesai membuat Workspace. Saat Anda membuka Workspace baru untuk pertama kalinya dan memilih panel kluster EMR, Anda akan melihat cluster yang Anda pilih terpasang.

In the Workspace UI

Melampirkan Workspace ke cluster komputasi Amazon EMR dari antarmuka pengguna Workspace

1. Di Workspace yang ingin Anda lampirkan ke cluster, pilih ikon cluster EMR dari sidebar kiri untuk membuka panel Cluster.
2. Di bawah tipe Cluster, perluas dropdown dan pilih EMR cluster pada EC2.
3. Pilih klaster dari daftar dropdown. Anda mungkin perlu melepaskan klaster yang ada terlebih dahulu untuk mengaktifkan daftar dropdown pilihan klaster.
4. Pilih Lampirkan. Ketika klaster terlampir, Anda akan melihat pesan berhasil muncul.

Melampirkan EMR Amazon di kluster EKS ke EMR Studio Workspace

Selain menggunakan klaster Amazon EMR yang berjalan di Amazon EC2, Anda dapat melampirkan Workspace untuk EMR Amazon pada klaster EKS untuk menjalankan kode notebook. Untuk informasi selengkapnya tentang Amazon EMR di EKS, lihat Apa itu Amazon EMR di EKS.

Sebelum dapat menghubungkan Workspace ke EMR Amazon di kluster EKS, administrator Studio harus memberi Anda izin akses.

> **Note**
>
> Anda tidak dapat meluncurkan EMR Amazon di kluster EKS di EMR Studio yang menggunakan propagasi identitas tepercaya IAM Identity Center.

On create

Untuk melampirkan EMR Amazon di kluster EKS saat Anda membuat Workspace

1. Dalam kotak dialog Create a Workspace, perluas bagian Konfigurasi lanjutan.
2. Pilih Lampirkan Workspace ke EMR Amazon di kluster EKS.
3. Di bawah Amazon EMR di kluster EKS, pilih cluster dari daftar dropdown.

4. Di bawah Pilih titik akhir, pilih endpoint terkelola untuk dilampirkan ke Workspace. Titik akhir terkelola adalah gateway yang memungkinkan EMR Studio berkomunikasi dengan klaster pilihan Anda.
5. Pilih Create a Workspace untuk menyelesaikan proses pembuatan Workspace dan lampirkan cluster yang dipilih.

Setelah melampirkan klaster, Anda dapat menyelesaikan proses pembuatan Workspace. Saat Anda membuka Workspace baru untuk pertama kalinya dan memilih panel kluster EMR, Anda akan melihat bahwa cluster yang Anda pilih terpasang.

In the Workspace UI

Untuk melampirkan EMR Amazon di kluster EKS dari antarmuka pengguna Workspace

1. Di Workspace yang ingin Anda lampirkan ke cluster, pilih ikon cluster EMR dari sidebar kiri untuk membuka panel Cluster.
2. Perluas dropdown tipe Cluster dan pilih cluster EMR di EKS.
3. Di bawah cluster EMR di EKS, pilih cluster dari daftar dropdown.
4. Di bawah Endpoint, pilih endpoint terkelola untuk dilampirkan ke Workspace. Titik akhir terkelola adalah gateway yang memungkinkan EMR Studio berkomunikasi dengan klaster pilihan Anda.
5. Pilih Lampirkan. Ketika klaster terlampir, Anda akan melihat pesan berhasil muncul.

Lampirkan aplikasi Amazon EMR Tanpa Server ke EMR Studio Workspace

Anda dapat melampirkan Workspace ke aplikasi EMR Serverless untuk menjalankan beban kerja interaktif. Untuk informasi selengkapnya, lihat Menggunakan notebook untuk menjalankan beban kerja interaktif dengan EMR Tanpa Server melalui EMR Studio.

> **Note**
>
> Anda tidak dapat melampirkan aplikasi EMR Tanpa Server ke EMR Studio yang menggunakan propagasi identitas tepercaya IAM Identity Center.

Example Lampirkan Workspace ke aplikasi EMR Serverless di JupyterLab

Sebelum Anda dapat menghubungkan Workspace ke aplikasi EMR Tanpa Server, administrator akun Anda harus memberi Anda izin akses seperti yang dijelaskan dalam Izin yang diperlukan untuk beban kerja interaktif.

1. Arahkan ke EMR Studio pilih Workspace Anda, lalu pilih Launch Workspace > Quick launch.
2. Di dalam JupyterLab, buka tab Cluster di sidebar kiri.
3. Pilih EMR Tanpa Server sebagai opsi komputasi, lalu pilih aplikasi EMR Tanpa Server dan peran runtime.
4. Untuk melampirkan cluster ke Workspace Anda, pilih Lampirkan.

Sekarang ketika Anda membuka Workspace ini, Anda akan melihat aplikasi yang Anda pilih terlampir.

Membuat dan melampirkan cluster EMR baru ke EMR Studio Workspace

Pengguna EMR Studio lanjutan dapat menyediakan klaster EMR baru yang berjalan di Amazon EC2 untuk digunakan dengan Workspace. Cluster baru memiliki semua aplikasi data besar yang diperlukan untuk EMR Studio diinstal secara default.

Untuk membuat klaster, administrator Studio Anda harus terlebih dahulu memberikan izin menggunakan kebijakan sesi. Untuk informasi selengkapnya, lihat Membuat kebijakan izin untuk pengguna EMR Studio.

Anda dapat membuat klaster baru di kotak dialog Buat Workspace atau dari panel Klaster di UI Workspace. Cara mana pun, Anda memiliki dua opsi pembuatan klaster:

1. Buat klaster EMR – Buat klaster EMR dengan memilih jenis dan jumlah instans Amazon EC2.
2. Gunakan template cluster — Menyediakan cluster dengan memilih template cluster yang telah ditentukan sebelumnya. Opsi ini muncul jika Anda memiliki izin untuk menggunakan template cluster.

> **Note**
>
> Jika Anda mengaktifkan propagasi identitas tepercaya dengan IAM Identity Center untuk Studio Anda, maka Anda harus menggunakan template untuk membuat klaster.

Untuk membuat klaster EMR dengan menyediakan konfigurasi klaster

1. Pilih titik awal.

Untuk…	Lakukan ini...
Membuat klaster saat Anda membuat Workspace dengan kotak dialog Buat Workspace.	Perluas bagian Konfigurasi lanjutan di kotak dialog Buat Workspace, dan pilih Buat klaster EMR.
Buat cluster dari panel cluster EMR di Workspace UI setelah Anda membuat Workspace.	Pilih tab EMR cluster di sidebar kiri Workspace terbuka, memperluas bagian Advanced configuration, dan pilih Create cluster.

2. Masukkan nama Nama kluster. Penamaan cluster membantu Anda menemukannya nanti di daftar EMR Studio Clusters.
3. Untuk rilis Amazon EMR, Pilih versi rilis Amazon EMR untuk cluster.
4. Misalnya, pilih jenis dan jumlah instans Amazon EC2 untuk cluster. Untuk informasi lebih lanjut tentang memilih jenis instans, lihat Konfigurasikan instans Amazon EC2. Satu contoh akan digunakan sebagai simpul utama.
5. Pilih Subnet tempat EMR Studio dapat meluncurkan cluster baru. Setiap opsi subnet telah disetujui sebelumnya oleh administrator Studio Anda, dan Workspace Anda harus dapat terhubung ke klaster di subnet apa pun yang terdaftar.
6. Pilih S3 URI untuk penyimpanan log.
7. Pilih Buat klaster EMR untuk menyediakan cluster. Jika Anda menggunakan kotak dialog Create a Workspace, pilih Create a Workspace untuk membuat Workspace dan menyediakan klaster. Setelah EMR Studio menyediakan cluster baru, ia melampirkan cluster ke Workspace.

Untuk membuat klaster menggunakan templat klaster

1. Pilih titik awal.

Untuk…	Lakukan ini...
Membuat klaster saat Anda membuat Workspace dengan kotak dialog Buat Workspace.	Perluas bagian Konfigurasi lanjutan di kotak dialog Buat ruang kerja, dan pilih Gunakan templat klaster.
Buat cluster dari panel cluster EMR di Workspace UI.	Pilih tab kluster EMR di bilah sisi kiri Workspace terbuka, perluas bagian Konfigurasi lanjutan, lalu pilih Template cluster.

2. Pilih templat klaster dari daftar dropdown. Setiap templat klaster yang tersedia mencakup deskripsi singkat untuk membantu Anda membuat pilihan.
3. Templat klaster yang Anda pilih mungkin memiliki parameter tambahan seperti versi rilis atau nama klaster Amazon EMR. Anda dapat memilih atau memasukkan nilai, atau menggunakan nilai default yang dipilih oleh administrator Anda.
4. Pilih Subnet tempat EMR Studio dapat meluncurkan cluster baru. Setiap opsi subnet telah disetujui sebelumnya oleh administrator Studio Anda, dan Workspace Anda harus dapat terhubung ke klaster di subnet apa pun.
5. Pilih Gunakan template cluster untuk menyediakan cluster dan melampirkannya ke Workspace. Ini akan memakan waktu beberapa menit bagi EMR Studio untuk membuat cluster. Jika Anda menggunakan kotak dialog Create a Workspace, pilih Create a Workspace untuk membuat Workspace dan menyediakan klaster. Setelah EMR Studio menyediakan cluster baru, ia melampirkan cluster ke Workspace Anda.

Lepaskan komputasi dari EMR Studio Workspace

Untuk menukar klaster yang dilampirkan ke Workspace, Anda dapat melepaskan cluster dari UI Workspace.

Untuk melepaskan cluster dari Workspace

1. Di Workspace yang ingin Anda lepaskan dari cluster, pilih ikon cluster EMR dari sidebar kiri untuk membuka panel Cluster.
2. Di bawah Pilih klaster, pilih Lepaskan dan tunggu EMR Studio melepaskan klaster tersebut. Ketika klaster terlepas, Anda akan melihat pesan sukses.

Untuk melepaskan aplikasi EMR Tanpa Server dari EMR Studio Workspace

Untuk menukar komputasi yang dilampirkan ke Workspace, Anda dapat melepaskan aplikasi dari UI Workspace.

1. Di Workspace yang ingin Anda lepaskan dari cluster, pilih ikon komputasi Amazon EMR dari bilah sisi kiri untuk membuka panel Compute.
2. Di bawah Pilih komputasi, pilih Lepaskan dan tunggu EMR Studio melepaskan aplikasi. Ketika aplikasi terlepas, Anda akan melihat pesan sukses.

Menautkan repositori berbasis Git ke Workspace EMR Studio

Tentang repositori Git untuk EMR Studio

Anda dapat mengaitkan maksimum tiga repositori Git dengan Workspace EMR Studio. Secara default, setiap Workspace memungkinkan Anda memilih dari daftar repositori Git yang terkait dengan AWS akun yang sama dengan Studio. Anda juga dapat membuat repositori Git baru sebagai sumber daya untuk Workspace.

Anda dapat menjalankan perintah Git seperti berikut menggunakan perintah terminal saat terhubung ke node utama dari sebuah cluster.

```
!git pull origin <branch-name>
```

Atau, Anda dapat menggunakan jupyterlab-git ekstensi. Buka dari bilah sisi kiri dengan memilih ikon Git. Untuk informasi tentang ekstensi jupyterlab-git untuk, lihat jupyterlab-git. JupyterLab

Prasyarat

- Untuk mengaitkan repositori Git dengan Workspace, Studio harus dikonfigurasi untuk mengizinkan penautan repositori Git. Administrator Studio Anda harus mengambil langkah-langkah untuk Membuat akses dan izin untuk repositori berbasis Git.
- Jika Anda menggunakan CodeCommit repositori, Anda harus menggunakan kredensi Git dan HTTPS. Kunci SSH dan HTTPS dengan pembantu AWS Command Line Interface kredensyal tidak didukung. CodeCommit juga tidak mendukung token akses pribadi (PATs). Untuk informasi selengkapnya, lihat Menggunakan IAM dengan CodeCommit Panduan pengguna IAM dan Pengaturan untuk pengguna HTTPS yang menggunakan kredensyal Git di Panduan Pengguna. AWS CodeCommit

Petunjuk

Untuk menautkan repositori Git terkait ke Workspace

1. Buka Workspace yang ingin Anda tautkan ke repositori dari daftar Workspaces di Studio.
2. Di bilah sisi kiri, pilih ikon Amazon EMR Git Repository untuk membuka panel alat repositori Git.
3. Di bawah repositori Git, perluas daftar dropdown dan pilih maksimal tiga repositori untuk ditautkan ke Workspace. EMR Studio mendaftarkan pilihan Anda dan mulai menautkan setiap repositori.

Mungkin perlu beberapa waktu hingga proses penautan selesai. Anda dapat melihat status untuk setiap repositori yang Anda pilih di panel alat Repositori Git. Setelah EMR Studio menautkan repositori ke Workspace, Anda akan melihat file milik repositori itu muncul di panel browser File.

Untuk menambahkan repositori Git baru ke Workspace sebagai sumber daya

1. Buka Workspace yang ingin Anda tautkan ke repositori dari daftar Workspace di Studio Anda.
2. Di bilah sisi kiri, pilih ikon Amazon EMR Git Repository untuk membuka panel alat repositori Git.
3. Pilih Tambahkan repositori Git baru.
4. Untuk Nama repositori, masukkan nama deskriptif untuk repositori di EMR Studio. Nama hanya boleh berisi karakter alfanumerik, tanda hubung, dan garis bawah.
5. Untuk URL repositori Git, masukkan URL untuk repositori. Ketika Anda menggunakan CodeCommit repositori, ini adalah URL yang disalin ketika Anda memilih Clone URL dan kemudian Clone HTTPS. Misalnya, `https://git-codecommit.us-west-2.amazonaws.com/v1/repos/[MyCodeCommitRepoName]`.
6. Untuk Cabang, masukkan nama cabang yang sudah ada yang ingin Anda periksa.
7. Untuk kredensial Git, pilih opsi sesuai dengan pedoman berikut. EMR Studio mengakses kredensial Git Anda menggunakan secret yang disimpan di Secrets Manager.

> **Note**
>
> Jika Anda menggunakan GitHub repositori, kami sarankan Anda menggunakan token akses pribadi (PAT) untuk mengautentikasi. Mulai 13 Agustus 2021, GitHub akan memerlukan otentikasi berbasis token dan tidak akan lagi menerima kata sandi saat

mengautentikasi operasi Git. Untuk informasi selengkapnya, lihat persyaratan otentikasi Token untuk posting operasi Git di GitHub Blog.

Opsi	Deskripsi
Buat secret baru	Pilih opsi ini untuk mengaitkan kredensial Git yang ada dengan secret baru yang akan dibuat di AWS Secrets Manager untuk Anda. Lakukan salah satu dari berikut ini berdasarkan kredensial Git yang Anda gunakan untuk repositori. Jika Anda menggunakan nama pengguna Git dan kata sandi untuk mengakses repositori, pilih Nama pengguna dan kata sandi, masukkan Nama secret untuk digunakan di Secrets Manager, kemudian masukkan Nama pengguna dan Kata Sandi untuk dikaitkan dengan secret. –ATAU– Jika Anda menggunakan token akses pribadi untuk mengakses repositori, pilih Token akses pribadi (PAT), masukkan Nama secret untuk digunakan di Secrets Manager, kemudian masukkan Token akses pribadi. Untuk informasi selengkapnya, lihat Membuat token akses pribadi untuk baris perintah GitHub dan Token akses pribadi untuk Bitbucket. CodeCommit repositori tidak mendukung opsi ini.
Gunakan repositori publik tanpa kredensial	Pilih opsi ini untuk mengakses repositori publik.

Opsi	Deskripsi
Gunakan secret AWS yang sudah ada	Pilih opsi ini jika Anda telah menyimpan kredensial Anda sebagai secret di Secrets Manager, lalu pilih nama secret dari daftar. Jika Anda memilih secret yang terkait dengan nama pengguna Git dan kata sandi, secret harus dalam format `{"gitUsername": "MyUserName ", "gitPassword": "MyPassword "}`.

8. Pilih Tambahkan repositori untuk membuat repositori baru. Setelah EMR Studio membuat repositori baru, Anda akan melihat pesan sukses. Repositori baru muncul dalam daftar dropdown di bawah Repositori Git.
9. Untuk menautkan repositori baru ke Workspace Anda, pilih dari daftar dropdown di bawah repositori Git.

Mungkin perlu beberapa waktu hingga proses penautan selesai. Setelah EMR Studio menautkan repositori baru ke Workspace, Anda akan melihat folder baru dengan nama yang sama dengan repositori Anda muncul di panel File Browser.

Untuk membuka repositori tertaut yang berbeda, arahkan ke foldernya di Peramban file.

Gunakan editor SQL Amazon Athena di EMR Studio

Gambaran Umum

Anda dapat menggunakan Amazon EMR Studio untuk mengembangkan dan menjalankan kueri interaktif di Amazon Athena. Itu berarti Anda dapat melakukan analisis SQL di Athena dari antarmuka EMR Studio yang sama yang Anda gunakan untuk menjalankan Spark, Scala, dan beban kerja lainnya. Dengan integrasi ini, Anda dapat menggunakan pelengkapan otomatis untuk mengembangkan kueri dengan cepat, menelusuri data di Katalog Data AWS Glue, membuat kueri yang disimpan, melihat riwayat kueri, dan banyak lagi.

Untuk informasi selengkapnya tentang penggunaan Amazon Athena, lihat Menggunakan Athena SQL di Panduan Pengguna Amazon Athena.

Gunakan editor SQL Athena di EMR Studio

Gunakan langkah-langkah berikut untuk mengembangkan dan menjalankan kueri interaktif di Amazon Athena dari EMR Studio Anda:

1. Tambahkan izin yang diperlukan ke peran pengguna untuk pengguna yang mengakses Ruang Kerja di Studio ini. Izin tercantum dalam AWS Identity and Access Managementizin untuk pengguna EMR Studio tabel di kolom Access Amazon Athena SQL editor dari EMR Studio Anda. Atau, Anda dapat memilih untuk menyalin konten kebijakan lanjutan dari Contoh kebijakan pengguna untuk memberikan pengguna izin penuh ke kemampuan EMR Studio termasuk yang satu ini.
2. Siapkan dan buat EMR Studio.
3. Arahkan ke Studio Anda dan pilih Editor kueri dari bilah sisi.

Anda sekarang harus melihat UI editor Athena yang sudah dikenal. Untuk informasi tentang memulai dan menggunakan Athena SQL untuk menjalankan kueri interaktif, lihat Memulai dan Menggunakan Athena SQL di Panduan Pengguna Amazon Athena.

> **Note**
>
> Jika Anda telah mengaktifkan propagasi identitas tepercaya melalui IAM Identity Center untuk EMR Studio Anda, maka Anda harus menggunakan workgroup Athena untuk mengontrol akses kueri, dan workgroup yang Anda gunakan juga harus menggunakan propagasi identitas tepercaya. Untuk langkah-langkah menyiapkan Pusat Identitas dan mengaktifkan propagasi identitas tepercaya untuk grup kerja Anda, lihat Menggunakan grup kerja Athena yang diaktifkan Pusat Identitas IAM di Panduan Pengguna Amazon Athena.

Pertimbangan untuk menggunakan editor Athena SQL di EMR Studio

- Integrasi dengan Athena tersedia di semua Wilayah komersial di mana EMR Studio dan Athena tersedia.
- Fitur Athena berikut tidak tersedia di EMR Studio:
  - Fitur admin seperti membuat atau memperbarui workgroup Athena, sumber data, atau reservasi kapasitas
  - Athena untuk notebook Spark atau Spark

- DataZone Integrasi Amazon
- Pengoptimal Berbasis Biaya (CBO)
- Fungsi langkah

CodeWhisperer Integrasi Amazon dengan EMR Studio Workspaces

Gambaran Umum

Anda dapat menggunakan Amazon CodeWhisperer dengan Amazon EMR Studio untuk mendapatkan rekomendasi waktu nyata saat Anda menulis kode. JupyterLab CodeWhispererdapat menyelesaikan komentar Anda, menyelesaikan satu baris kode, membuat line-by-line rekomendasi, dan menghasilkan fungsi yang sepenuhnya terbentuk.

> **Note**
>
> Saat Anda menggunakan Amazon EMR Studio, AWS mungkin menyimpan data tentang penggunaan dan konten Anda untuk tujuan peningkatan layanan. Untuk informasi selengkapnya dan petunjuk untuk memilih keluar dari berbagi data, lihat Berbagi data Anda AWS di Panduan CodeWhisperer Pengguna Amazon.

Pertimbangan untuk digunakan CodeWhisperer dengan Ruang Kerja

- CodeWhisperer integrasi tersedia di Wilayah AWS tempat yang sama di mana EMR Studio tersedia, seperti yang didokumentasikan dalam pertimbangan EMR Studio.
- Amazon EMR Studio secara otomatis menggunakan CodeWhisperer endpoint di US East (Virginia N.) (us-east-1) untuk rekomendasi, terlepas dari Wilayah tempat studio Anda berada.
- CodeWhisperer hanya mendukung bahasa Python untuk pengkodean skrip ETL untuk pekerjaan Spark di EMR Studio.
- Opsi telemetri sisi klien mengukur penggunaan Anda. CodeWhisperer Fungsionalitas ini tidak didukung dengan EMR Studio.

Izin diperlukan untuk CodeWhisperer

Untuk menggunakannya CodeWhisperer, Anda harus melampirkan kebijakan berikut ke peran pengguna IAM Anda untuk Amazon EMR Studio:

```
{
    "Version": "2012-10-17",
    "Statement": [
        {
          "Sid": "CodeWhispererPermissions",
          "Effect": "Allow",
          "Action": [ "codewhisperer:GenerateRecommendations" ],
          "Resource": "*"
        }
     ]
}
```

Gunakan CodeWhisperer dengan Ruang Kerja

Untuk menampilkan log CodeWhisperer referensi JupyterLab, buka CodeWhispererpanel di bagian bawah JupyterLab jendela dan pilih Buka Log Referensi Kode.

Daftar berikut berisi pintasan yang dapat Anda gunakan untuk berinteraksi dengan CodeWhisperer saran:

- Rekomendasi jeda — Gunakan Jeda Saran Otomatis dari pengaturan. CodeWhisperer
- Terima rekomendasi — Tekan Tab pada keyboard Anda.
- Tolak rekomendasi — Tekan Escape pada keyboard Anda.
- Navigasi rekomendasi — Gunakan panah Atas dan Bawah pada keyboard Anda.
- Pemanggilan manual — Tekan Alt dan C pada keyboard Anda. Jika Anda menggunakan Mac, tekan Cmd dan C.

Anda juga dapat menggunakan CodeWhisperer untuk mengubah pengaturan seperti tingkat log dan mendapatkan saran untuk referensi kode. Untuk informasi selengkapnya, lihat Menyiapkan CodeWhisperer dengan JupyterLab dan Fitur di Panduan CodeWhisperer Pengguna Amazon.

Debug aplikasi dan pekerjaan dengan EMR Studio

Dengan Amazon EMR Studio, Anda dapat meluncurkan antarmuka aplikasi data untuk menganalisis aplikasi dan pekerjaan yang berjalan di browser.

Anda juga dapat meluncurkan antarmuka pengguna yang persisten dan di luar klaster untuk Amazon EMR yang berjalan pada klaster EC2 dari konsol Amazon EMR. Untuk informasi selengkapnya, lihat Melihat antarmuka pengguna aplikasi persisten.

> **Note**
>
> Bergantung pada setelan peramban, Anda mungkin perlu mengaktifkan pop-up agar UI aplikasi terbuka.

Untuk informasi tentang mengonfigurasi dan menggunakan antarmuka aplikasi, lihat Server Timeline YARN, Pemantauan dan instrumentasi, atau Gambaran umum Tez UI.

Men-debug Amazon EMR yang berjalan pada pekerjaan Amazon EC2

Workspace UI

Luncurkan UI pada klaster dari file notebook

Jika Anda menggunakan rilis Amazon EMR versi 5.33.0 dan yang lebih baru, Anda dapat meluncurkan antarmuka pengguna web Spark (Spark UI atau Spark History Server) dari notebook di Workspace Anda.

UI on-cluster bekerja dengan kernel PySpark, Spark, atau SparkR. Ukuran maksimum file dapat dilihat untuk log peristiwa atau log kontainer Spark adalah 10 MB. Jika file log melebihi 10 MB, sebaiknya Anda menggunakan Spark History Server yang persisten, bukannya Spark UI pada klaster untuk men-debug pekerjaan.

> **Important**
>
> Agar EMR Studio dapat meluncurkan antarmuka pengguna aplikasi on-cluster dari Workspace, kluster harus dapat berkomunikasi dengan Amazon API Gateway. Anda harus mengonfigurasi kluster EMR untuk mengizinkan lalu lintas jaringan keluar ke Amazon API Gateway, dan memastikan bahwa Amazon API Gateway dapat dijangkau dari cluster.
> Spark UI mengakses log kontainer dengan menyelesaikan nama host. Jika Anda menggunakan nama domain khusus, Anda harus memastikan bahwa nama host simpul klaster Anda dapat diselesaikan oleh Amazon DNS atau server DNS yang Anda tentukan. Untuk melakukannya, atur opsi Dynamic Host Configuration Protocol (DHCP) untuk Amazon Virtual Private Cloud (VPC) yang terkait dengan klaster Anda. Untuk informasi lebih lanjut tentang opsi DHCP, lihat Set opsi DHCP dalam Panduan Pengguna Amazon Virtual Private Cloud.

1. Di EMR Studio Anda, buka Workspace yang ingin Anda gunakan dan pastikan itu terlampir ke klaster Amazon EMR yang berjalan di EC2. Untuk instruksi, lihat Lampirkan komputasi ke Ruang Kerja EMR Studio.
2. Buka file notebook dan gunakan kernel PySpark, Spark, atau SparkR. Untuk memilih kernel, pilih nama kernel dari kanan atas bilah alat notebook untuk membuka kotak dialog Pilih Kernel. Nama muncul sebagai Tidak ada Kernel! jika tidak ada kernel yang dipilih.
3. Jalankan kode notebook Anda. Berikut ini muncul sebagai output di notebook ketika Anda memulai konteks Spark. Mungkin diperlukan waktu beberapa detik untuk muncul. Jika Anda telah memulai konteks Spark, Anda dapat menjalankan `%%info` perintah untuk mengakses tautan ke UI Spark kapan saja.

> **Note**
>
> Jika tautan Spark UI tidak berfungsi atau tidak muncul setelah beberapa detik, buat sel notebook baru dan jalankan perintah `%%info` untuk meregenerasi tautan.

4. Untuk meluncurkan Spark UI, pilih Tautan di bawah Spark UI. Jika aplikasi Spark Anda sedang berjalan, Spark UI terbuka di tab baru. Jika aplikasi telah selesai, Spark History Server akan membuka.

   Setelah meluncurkan UI Spark, Anda dapat memodifikasi URL di browser untuk membuka YARN ResourceManager atau Yarn Timeline Server. Tambahkan salah satu jalur berikut setelah `amazonaws.com`.

Web UI	Jalur	Contoh URL yang dimodifikasi
BENANG ResourceM anager	/rm	https://*j-examplebby5ij* .emrappui-prod.*eu-west-1* .amazonaws.com*/rm*
Yarn Timeline Server	/yts	https://*j-examplebby5ij* .emrappui-prod.*eu-west-1* .amazonaws.com*/yts*
Spark History Server	/shs	https://*j-examplebby5ij* .emrappui-prod.*eu-west-1* .amazonaws.com*/shs*

Studio UI

Luncurkan YARN Timeline Server, Spark History Server, atau Tez UI persisten dari EMR Studio UI

1. Di EMR Studio Anda, pilih Amazon EMR di EC2 di sisi kiri halaman untuk membuka Amazon EMR pada daftar cluster EC2.
2. Filter daftar klaster menurut nama, status, atau ID dengan memasukkan nilai di kotak pencarian. Anda juga dapat mencari berdasarkan rentang waktu pembuatan.
3. Pilih klaster kemudian pilih Luncurkan UI aplikasi untuk memilih antarmuka pengguna aplikasi. UI Aplikasi terbuka di tab peramban baru dan mungkin memerlukan beberapa waktu untuk memuat.

Debug EMR Studio berjalan di EMR Tanpa Server

Mirip dengan Amazon EMR yang berjalan di Amazon EC2, Anda dapat menggunakan antarmuka pengguna Workspace untuk menganalisis aplikasi EMR Tanpa Server Anda. Dari UI Workspace, saat Anda menggunakan Amazon EMR rilis 6.14.0 dan yang lebih tinggi, Anda dapat meluncurkan antarmuka pengguna web Spark (UI Spark atau Server Riwayat Spark) dari notebook di Workspace Anda. Untuk kenyamanan Anda, kami juga menyediakan tautan ke log driver untuk akses cepat log driver Spark.

Debug Amazon EMR pada pekerjaan EKS berjalan dengan Spark History Server

Saat Anda mengirimkan pekerjaan yang dijalankan ke EMR Amazon di klaster EKS, Anda dapat mengakses log untuk pekerjaan yang dijalankan menggunakan Server Riwayat Spark. Spark History Server menyediakan alat untuk memantau aplikasi Spark, seperti daftar tahapan dan tugas penjadwal, ringkasan ukuran RDD dan penggunaan memori, dan informasi lingkungan. Anda dapat meluncurkan Spark History Server untuk Amazon EMR pada pekerjaan EKS berjalan dengan cara berikut:

- Saat mengirimkan pekerjaan yang dijalankan menggunakan EMR Studio dengan Amazon EMR di titik akhir terkelola EKS, Anda dapat meluncurkan Server Riwayat Spark dari file notebook di Workspace.
- Saat Anda mengirimkan pekerjaan yang dijalankan menggunakan AWS CLI atau AWS SDK untuk Amazon EMR di EKS, Anda dapat meluncurkan Spark History Server dari EMR Studio UI.

Untuk informasi tentang cara menggunakan Spark History Server, lihat Pemantauan dan Instrumentasi dalam dokumentasi Apache Spark. Untuk informasi lebih lanjut tentang pekerjaan berjalan, lihat Konsep dan komponen dalam Panduan Pengembangan Amazon EMR pada EKS.

Untuk meluncurkan Spark History Server dari file notebook di EMR Studio Workspace

1. Buka Workspace yang terhubung ke Amazon EMR di klaster EKS.
2. Pilih dan buka file notebook Anda di Workspace.
3. Pilih Spark UI di bagian atas file notebook untuk membuka Server Riwayat Spark persisten di tab baru.

Untuk meluncurkan Spark History Server dari EMR Studio UI

> **Note**
>
> Daftar Pekerjaan di EMR Studio UI hanya menampilkan tugas yang Anda kirimkan menggunakan AWS CLI atau AWS SDK untuk Amazon EMR di EKS.

1. Di EMR Studio Anda, pilih Amazon EMR di EKS di sisi kiri halaman.

2. Cari EMR Amazon di klaster virtual EKS yang Anda gunakan untuk mengirimkan pekerjaan Anda. Anda dapat memfilter daftar cluster berdasarkan status atau ID dengan memasukkan nilai di kotak pencarian.
3. Pilih cluster untuk membuka halaman detailnya. Halaman detail menampilkan informasi tentang cluster, seperti ID, namespace, dan status. Halaman ini juga menampilkan daftar semua pekerjaan yang dikirimkan ke klaster itu.
4. Dari halaman detail klaster, pilih pekerjaan berjalan untuk di-debug.
5. Di kanan atas daftar Pekerjaan, pilih Luncurkan Spark History Server untuk membuka antarmuka aplikasi di tab peramban baru.

Instal kernel dan pustaka di Ruang Kerja EMR Studio

Setiap Amazon EMR Studio Workspace dilengkapi dengan serangkaian pustaka dan kernel yang sudah diinstal sebelumnya.

Kernel dan pustaka pada cluster yang berjalan di Amazon EC2

Anda juga dapat menyesuaikan lingkungan untuk EMR Studio dengan cara berikut ketika Anda menggunakan klaster EMR yang berjalan di Amazon EC2:

- Instal kernel Jupyter Notebook dan pustaka Python pada simpul utama klaster — Saat Anda menginstal pustaka menggunakan opsi ini, semua Ruang Kerja yang dilampirkan ke klaster yang sama berbagi pustaka tersebut. Anda dapat menginstal kernel atau pustaka dari dalam sel notebook atau saat terhubung menggunakan SSH ke node utama cluster.
- Gunakan pustaka dengan cakupan notebook — Saat pengguna Workspace menginstal dan menggunakan pustaka dari dalam sel notebook, pustaka tersebut hanya tersedia untuk buku catatan itu saja. Opsi ini memungkinkan notebook yang berbeda menggunakan klaster yang sama bekerja tanpa khawatir tentang versi pustaka yang bertentangan.

EMR Studio Workspaces memiliki arsitektur dasar yang sama dengan EMR Notebooks. Anda dapat menginstal dan menggunakan kernel Notebook Jupyter dan pustaka Python dengan EMR Studio dengan cara yang sama seperti yang Anda lakukan dengan EMR Notebooks. Untuk instruksi, lihat Memasang dan menggunakan kernel dan pustaka.

Kernel dan pustaka di Amazon EMR pada klaster EKS

Amazon EMR pada kluster EKS menyertakan kernel dan PySpark Python 3.7 dengan satu set pustaka yang sudah diinstal sebelumnya. Amazon EMR di EKS tidak mendukung pemasangan pustaka atau cluster tambahan.

Setiap Amazon EMR di kluster EKS dilengkapi dengan Python dan pustaka berikut yang diinstal: PySpark

- Python – boto3, cffi, future, ggplot, jupyter, kubernetes, matplotlib, numpy, pandas, plotly, pycryptodomex, py4j, requests, scikit-learn, scipy, seaborn
- PySpark – ggplot, jupyter, matplotlib, numpy, pandas, plotly, pycryptodomex, py4j, requests, scikit-learn, scipy, seaborn

Kernel dan pustaka pada aplikasi EMR Tanpa Server

Setiap aplikasi EMR Tanpa Server dilengkapi dengan Python dan pustaka berikut yang diinstal: PySpark

- Python – ggplot, matplotlib, numpy, pandas, plotly, bokeh, scikit-learn, scipy, seaborn
- PySpark – ggplot, matplotlib,numpy, pandas, plotly, bokeh, scikit-learn, scipy, seaborn

Tingkatkan kernel dengan perintah magic

Gambaran Umum

EMR Studio dan EMR Notebooks mendukung perintah. magic Magicperintah, atau magics, adalah perangkat tambahan yang disediakan kernel IPython untuk membantu Anda menjalankan dan menganalisis data. IPython adalah lingkungan shell interaktif yang dibangun dengan Python.

Amazon EMR juga mendukungSparkmagic, paket yang menyediakan kernel terkait Spark (PySpark, SparkR, dan kernel Scala) dengan perintah magic tertentu dan yang menggunakan Livy di cluster untuk mengirimkan pekerjaan Spark.

Anda dapat menggunakan magic perintah selama Anda memiliki kernel Python di notebook EMR Anda. Demikian pula, kernel terkait SPARK apa pun mendukung Sparkmagic perintah.

Magicperintah, juga disebut magics, datang dalam dua varietas:

- Baris magic s — magic Perintah ini dilambangkan dengan `%` awalan tunggal dan beroperasi pada satu baris kode
- Sel magic s — magic Perintah ini dilambangkan dengan `%%` awalan ganda dan beroperasi pada beberapa baris kode

Untuk semua magic s yang tersedia, lihatDaftar magic dan Sparkmagic perintah.

Pertimbangan dan batasan

- EMR Tanpa Server tidak mendukung untuk dijalankan. `%%sh spark-submit` Itu tidak mendukung EMR magic Notebooks s.
- Amazon EMR di kluster EKS tidak mendukung perintah Sparkmagic untuk EMR Studio. Ini karena kernel Spark yang Anda gunakan dengan endpoint terkelola dibangun ke dalam Kubernetes, dan kernel tersebut tidak didukung oleh dan Livy. Sparkmagic Anda dapat mengatur konfigurasi Spark langsung ke SparkContext objek sebagai solusi, seperti yang ditunjukkan oleh contoh berikut.

  ```
  spark.conf.set("spark.driver.maxResultSize", '6g')
  ```

- magicPerintah dan tindakan berikut dilarang olehAWS:
  - `%alias`
  - `%alias_magic`
  - `%automagic`
  - `%macro`
  - Memodifikasi dengan `proxy_user` `%configure`
  - Memodifikasi `KERNEL_USERNAME` dengan `%env` atau `%set_env`

Daftar magic dan Sparkmagic perintah

Gunakan perintah berikut untuk membuat daftar magic perintah yang tersedia:

- `%lsmagic`daftar semua fungsi yang tersedia saat ini. magic
- `%%help`mencantumkan magic fungsi terkait SPARK yang tersedia saat ini yang disediakan oleh paket. Sparkmagic

Gunakan `%%configure` untuk mengkonfigurasi Spark

Salah satu perintah yang paling berguna adalah Sparkmagic `%%configure` perintah, yang mengkonfigurasi parameter pembuatan sesi. Menggunakan `conf` pengaturan, Anda dapat mengonfigurasi konfigurasi Spark apa pun yang disebutkan dalam dokumentasi konfigurasi untuk Apache Spark.

Example Tambahkan file JAR eksternal ke EMR Notebooks dari repositori Maven atau Amazon S3

Anda dapat menggunakan pendekatan berikut untuk menambahkan dependensi file JAR eksternal ke kernel terkait SPARK yang didukung oleh. Sparkmagic

```
%%configure -f
{"conf": {
    "spark.jars.packages": "com.jsuereth:scala-arm_2.11:2.0,ml.combust.bundle:bundle-
ml_2.11:0.13.0,com.databricks:dbutils-api_2.11:0.0.3",
    "spark.jars": "s3://DOC-EXAMPLE-BUCKET/my-jar.jar"
    }
}
```

Example : Konfigurasikan Hudi

Anda dapat menggunakan editor notebook untuk mengonfigurasi notebook EMR Anda untuk menggunakan Hudi.

```
%%configure
{ "conf": {
     "spark.jars": "hdfs://apps/hudi/lib/hudi-spark-bundle.jar,hdfs:///apps/hudi/lib/
spark-spark-avro.jar",
     "spark.serializer": "org.apache.spark.serializer.KryoSerializer",
     "spark.sql.hive.convertMetastoreParquet":"false"
     }
}
```

Gunakan `%%sh` untuk menjalankan `spark-submit`

`%%sh`magicMenjalankan perintah shell dalam subproses pada instance cluster terlampir Anda. Biasanya, Anda akan menggunakan salah satu kernel terkait Spark untuk menjalankan aplikasi Spark pada cluster terlampir Anda. Namun, jika Anda ingin menggunakan kernel Python untuk mengirimkan aplikasi Spark, Anda dapat menggunakan yang berikut inimagic, mengganti nama bucket dengan nama bucket Anda dalam huruf kecil.

```
%%sh
spark-submit --master yarn --deploy-mode cluster s3://DOC-EXAMPLE-BUCKET/test.py
```

Dalam contoh ini, cluster membutuhkan akses ke lokasis3://*DOC-EXAMPLE-BUCKET*/test.py, atau perintah akan gagal.

Anda dapat menggunakan perintah Linux apa pun dengan file %%shmagic. Jika Anda ingin menjalankan perintah Spark atau YARN, gunakan salah satu opsi berikut untuk membuat pengguna emr-notebook Hadoop dan berikan izin pengguna untuk menjalankan perintah:

- Anda dapat secara eksplisit membuat pengguna baru dengan menjalankan perintah berikut.

  ```
  hadoop fs -mkdir /user/emr-notebook
  hadoop fs -chown emr-notebook /user/emr-notebook
  ```

- Anda dapat mengaktifkan peniruan identitas pengguna di Livy, yang secara otomatis membuat pengguna. Lihat Mengaktifkan peniruan pengguna untuk memantau aktivitas pengguna dan tugas Spark untuk informasi selengkapnya.

Gunakan **%%display** untuk memvisualisasikan kerangka data Spark

Anda dapat menggunakan %%display magic untuk memvisualisasikan kerangka data Spark. Untuk menggunakan inimagic, jalankan perintah berikut.

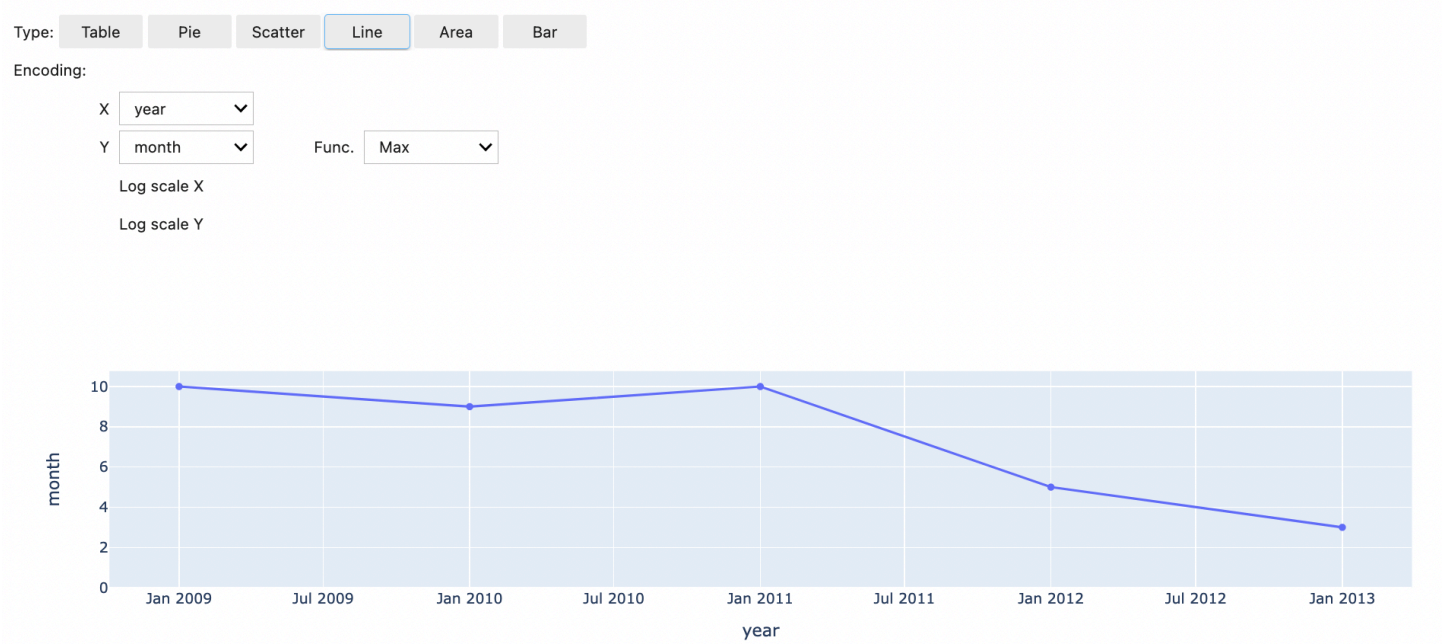
```
%%display df
```

Pilih untuk melihat hasil dalam format tabel, seperti yang ditunjukkan gambar berikut.

Type: Table Pie Scatter Line Area Bar

year	month	total_passengers	total_trips
2012-01-01	3	26866837	16146923
2011-01-01	3	26091246	16066350
2013-01-01	3	26965079	15749228
2011-01-01	10	26287953	15707756
2009-01-01	10	26202049	15604551
2012-01-01	5	26278817	15567525
2011-01-01	5	25508952	15554868
2010-01-01	9	25533166	15540209
2010-01-01	5	26002858	15481351
2012-01-01	4	25900645	15477914

Anda juga dapat memilih untuk memvisualisasikan data Anda dengan lima jenis bagan. Pilihan Anda termasuk diagram pie, scatter, line, area, dan bar.



Gunakan magic EMR Notebooks s

Amazon EMR menyediakan EMR Notebooks berikut yang dapat Anda gunakan dengan magic kernel berbasis Python3 dan Spark:

- `%mount_workspace_dir`- Memasang direktori Workspace Anda ke cluster Anda sehingga Anda dapat mengimpor dan menjalankan kode dari file lain di Workspace Anda

> **Note**
>
> Dengan`%mount_workspace_dir`, hanya kernel Python 3 yang dapat mengakses sistem file lokal Anda. Eksekutor Spark tidak akan memiliki akses ke direktori yang dipasang dengan kernel ini.

- `%umount_workspace_dir`- Melepas direktori Workspace Anda dari cluster Anda
- `%generate_s3_download_url`- Menghasilkan tautan unduhan sementara di output notebook Anda untuk objek Amazon S3

Prasyarat

Sebelum Anda menginstal EMR magic Notebooks s, selesaikan tugas-tugas berikut:

- Pastikan Anda Peran layanan untuk instans EC2 klaster (profil instans EC2) memiliki akses baca untuk Amazon S3. `EMR_EC2_DefaultRole`Dengan kebijakan yang `AmazonElasticMapReduceforEC2Role` dikelola memenuhi persyaratan ini. Jika Anda menggunakan peran atau kebijakan khusus, pastikan bahwa itu memiliki izin S3 yang diperlukan.

> **Note**
>
> EMR magic Notebooks berjalan di cluster sebagai pengguna notebook dan menggunakan profil instans EC2 untuk berinteraksi dengan Amazon S3. Saat Anda memasang direktori Workspace pada kluster EMR, semua Workspaces dan notebook EMR dengan izin untuk melampirkan ke cluster tersebut dapat mengakses direktori yang dipasang. Direktori dipasang sebagai read-only secara default. Sementara `s3fs-fuse` dan `goofys` mengizinkan pemasangan baca-tulis, kami sangat menyarankan agar Anda tidak memodifikasi parameter pemasangan untuk memasang direktori dalam mode baca-tulis. Jika Anda mengizinkan akses tulis, setiap perubahan yang dilakukan pada direktori ditulis ke bucket S3. Untuk menghindari penghapusan atau penimpaan yang tidak disengaja,

Anda dapat mengaktifkan pembuatan versi untuk bucket S3 Anda. Untuk mempelajari lebih lanjut, lihat Menggunakan pembuatan versi di bucket S3.

- Jalankan salah satu skrip berikut di cluster Anda untuk menginstal dependensi untuk EMR Notebooks s. magic Untuk menjalankan skrip, Anda dapat Gunakan tindakan bootstrap kustom atau mengikuti instruksi dalam perintah Jalankan dan skrip di klaster EMR Amazon saat Anda sudah memiliki cluster yang sedang berjalan.

  Anda dapat memilih dependensi mana yang akan diinstal. Baik s3fs-fuse dan goofys adalah alat FUSE (Filesystem in Userspace) yang memungkinkan Anda memasang bucket Amazon S3 sebagai sistem file lokal di cluster. `s3fs`Alat ini memberikan pengalaman yang mirip dengan POSIX. `goofys`Alat ini adalah pilihan yang baik ketika Anda lebih memilih kinerja daripada sistem file yang sesuai dengan POSIX.

```
#!/bin/sh

# Install the s3fs dependency for EMR Notebooks magics sudo amazon-linux-extras
 install epel -y
sudo yum install s3fs-fuse -y
```

  ATAU

```
#!/bin/sh

# Install the goofys dependency for EMR Notebooks magics sudo wget https://
github.com/kahing/goofys/releases/latest/download/goofys -P /usr/bin/
sudo chmod ugo+x /usr/bin/goofys
```

Instal magic EMR Notebooks s

> **Note**
>
> Dengan Amazon EMR merilis 6.0 hingga 6.9.0, dan 5.0 hingga 5.36.0, hanya versi paket 0.2.0 dan dukungan yang lebih tinggi. `emr-notebooks-magics` `%mount_workspace_dir` magic

Selesaikan langkah-langkah berikut untuk menginstal EMR Notebooks smagic.

1. Di notebook Anda, jalankan perintah berikut untuk menginstal [emr-notebooks-magics](#)paket.

```
%pip install boto3 --upgrade
%pip install botocore --upgrade
%pip install emr-notebooks-magics --upgrade
```

2. Mulai ulang kernel Anda untuk memuat EMR magic Notebooks s.
3. Verifikasi instalasi Anda dengan perintah berikut, yang akan menampilkan teks bantuan output untuk`%mount_workspace_dir`.

```
%mount_workspace_dir?
```

Pasang direktori Workspace dengan **`%mount_workspace_dir`**

Ini `%mount_workspace_dir` magic memungkinkan Anda memasang direktori Workspace ke kluster EMR sehingga Anda dapat mengimpor dan menjalankan file, modul, atau paket lain yang disimpan di direktori Anda.

Contoh berikut memasang seluruh direktori Workspace ke cluster, dan menentukan *<--fuse-type>*argumen opsional untuk menggunakan goofys untuk memasang direktori.

```
%mount_workspace_dir . <--fuse-type goofys>
```

Untuk memverifikasi bahwa direktori Workspace Anda sudah terpasang, gunakan contoh berikut untuk menampilkan direktori kerja saat ini dengan `ls` perintah. Output harus menampilkan semua file di Workspace Anda.

```
%%sh
ls
```

Setelah selesai membuat perubahan di Workspace, Anda dapat melepas direktori Workspace dengan perintah berikut:

> **Note**
>
> Direktori Workspace Anda tetap terpasang ke klaster Anda bahkan ketika Workspace dihentikan atau terlepas. Anda harus secara eksplisit melepas direktori Workspace Anda.

```
%umount_workspace_dir
```

Unduh objek Amazon S3 dengan **`%generate_s3_download_url`**

`generate_s3_download_url`Perintah membuat URL presigned untuk objek yang disimpan di Amazon S3. Anda dapat menggunakan URL yang telah ditetapkan sebelumnya untuk mengunduh objek ke mesin lokal Anda. Misalnya, Anda mungkin menjalankan `generate_s3_download_url` untuk mengunduh hasil kueri SQL yang ditulis kode Anda ke Amazon S3.

URL presigned valid selama 60 menit secara default. Anda dapat mengubah waktu kedaluwarsa dengan menentukan beberapa detik untuk bendera. `--expires-in` Misalnya, `--expires-in 1800` membuat URL yang valid selama 30 menit.

Contoh berikut menghasilkan tautan unduhan untuk objek dengan menentukan jalur Amazon S3 lengkap:. *`s3://EXAMPLE-DOC-BUCKET/path/to/my/object`*

```
%generate_s3_download_url s3://EXAMPLE-DOC-BUCKET/path/to/my/object
```

Untuk mempelajari lebih lanjut tentang menggunakan`generate_s3_download_url`, jalankan perintah berikut untuk menampilkan teks bantuan.

```
%generate_s3_download_url?
```

Jalankan notebook dalam mode tanpa kepala dengan **`%execute_notebook`**

Dengan `%execute_notebook`magic, Anda dapat menjalankan notebook lain dalam mode headless dan melihat output untuk setiap sel yang telah Anda jalankan. Ini magic memerlukan izin tambahan untuk peran instans yang dibagikan Amazon EMR dan Amazon EC2. Untuk detail selengkapnya tentang cara memberikan izin tambahan, jalankan perintah`%execute_notebook?`.

Selama pekerjaan yang berjalan lama, sistem Anda mungkin tertidur karena tidak aktif, atau mungkin kehilangan konektivitas internet untuk sementara. Ini mungkin mengganggu koneksi antara browser Anda dan Server Jupyter. Dalam hal ini, Anda mungkin kehilangan output dari sel yang telah Anda jalankan dan kirim dari Server Jupyter.

Jika Anda menjalankan notebook dalam mode headless dengan `%execute_notebook`magic, EMR Notebooks menangkap output dari sel yang telah berjalan, bahkan jika jaringan lokal mengalami gangguan. EMR Notebooks menyimpan output secara bertahap di notebook baru dengan nama

yang sama dengan notebook yang Anda jalankan. EMR Notebooks kemudian menempatkan notebook ke folder baru di dalam ruang kerja. Proses tanpa kepala terjadi pada cluster yang sama dan menggunakan peran layananEMR_Notebook_DefaultRole, tetapi argumen tambahan dapat mengubah nilai default.

Untuk menjalankan notebook dalam mode headless, gunakan perintah berikut:

```
%execute_notebook <relative-file-path>
```

Untuk menentukan ID klaster dan peran layanan untuk menjalankan tanpa kepala, gunakan perintah berikut:

```
%execute_notebook <notebook_name>.ipynb --cluster-id <emr-cluster-id> --service-role
 <emr-notebook-service-role>
```

Saat Amazon EMR dan Amazon EC2 berbagi peran instans, peran tersebut memerlukan izin tambahan berikut:

```
{
    "Version": "2012-10-17",
    "Statement": [
        {
            "Effect": "Allow",
            "Action": [
                "elasticmapreduce:StartNotebookExecution",
                "elasticmapreduce:DescribeNotebookExecution",
                "ec2:DescribeInstances"
            ],
            "Resource": "*"
        },
        {
            "Effect": "Allow",
            "Action": [
                "iam:PassRole"
            ],
            "Resource": "arn:aws:iam::<AccoundId>:role/EMR_Notebooks_DefaultRole"
        }
    ]
}
```

> **Note**
>
> Untuk menggunakan `%execute_notebook`magic, instal `emr-notebooks-magics` paket, versi 0.2.3 atau lebih tinggi.

Gunakan notebook multi-bahasa dengan kernel Spark

Setiap kernel notebook Jupyter memiliki bahasa default. Misalnya, bahasa default kernel Spark adalah Scala, dan bahasa default PySpark kernel adalah Python. Dengan Amazon EMR 6.4.0 dan yang lebih baru, EMR Studio mendukung notebook multi-bahasa. Ini berarti bahwa setiap kernel di EMR Studio dapat mendukung bahasa berikut selain bahasa default: Python, Spark, R, dan Spark SQL.

Untuk mengaktifkan fitur ini, tentukan salah satu magic perintah berikut di awal sel apa pun.

Bahasa	Perintah
Python	`%%pyspark`
Skala	`%%scalaspark`
R	`%%rspark` Tidak didukung untuk beban kerja interaktif dengan EMR Tanpa Server.
Spark SQL	`%%sql`

Saat dipanggil, perintah ini menjalankan seluruh sel dalam sesi Spark yang sama menggunakan penerjemah bahasa yang sesuai.

`%%pyspark`Sel magic memungkinkan pengguna untuk menulis PySpark kode di semua kernel Spark.

```
%%pyspark
a = 1
```

`%%sql`Sel magic memungkinkan pengguna untuk mengeksekusi kode Spark-SQL di semua kernel Spark.

```
%%sql
SHOW TABLES
```

`%%rspark`Sel magic memungkinkan pengguna untuk mengeksekusi kode SparkR di semua kernel Spark.

```
%%rspark
a <- 1
```

`%%scalaspark`Sel magic memungkinkan pengguna untuk mengeksekusi kode Spark Scala di semua kernel Spark.

```
%%scalaspark
val a = 1
```

Bagikan data di seluruh penerjemah bahasa dengan tabel sementara

Anda juga dapat berbagi data antar penerjemah bahasa menggunakan tabel sementara. Contoh berikut menggunakan `%%pyspark` dalam satu sel untuk membuat tabel sementara di Python dan menggunakan `%%scalaspark` dalam sel berikut untuk membaca data dari tabel itu di Scala.

```
%%pyspark
df=spark.sql("SELECT * from nyc_top_trips_report LIMIT 20")
# create a temporary table called nyc_top_trips_report_view in python
df.createOrReplaceTempView("nyc_top_trips_report_view")
```

```
%%scalaspark
// read the temp table in scala
val df=spark.sql("SELECT * from nyc_top_trips_report_view")
df.show(5)
```

Ikhtisar Amazon EMR Notebooks

> **Note**
>
> EMR Notebooks tersedia sebagai EMR Studio Workspaces di konsol baru. Anda masih dapat menggunakan notebook yang ada di konsol lama, tetapi Anda tidak dapat membuat notebook baru di konsol lama. Tombol Create Workspace di konsol baru menggantikan fungsi ini. Untuk mengakses atau membuat Ruang Kerja, pengguna EMR Notebooks memerlukan izin peran IAM tambahan. Untuk informasi selengkapnya, lihat Amazon EMR Notebook adalah Amazon EMR Studio Workspaces di konsol baru dan Apa yang baru di konsol?

Anda dapat menggunakan Amazon EMR Notebooks bersama dengan Amazon EMR cluster yang menjalankan Apache Spark untuk membuat dan membuka Jupyter Notebook dan antarmuka dalam konsol Amazon EMR. JupyterLab Notebook EMR adalah notebook "nirserver" yang dapat Anda gunakan untuk menjalankan kueri dan kode. Tidak seperti notebook tradisional, isi notebook EMR — persamaan, kueri, model, kode, dan teks naratif dalam sel notebook — berjalan di klien. Perintah dijalankan menggunakan kernel pada klaster EMR. Isi notebook juga disimpan ke Amazon S3 secara terpisah dari data klaster untuk daya tahan dan penggunaan kembali yang fleksibel.

Anda dapat memulai sebuah klaster, melampirkan notebook EMR untuk analisis, dan kemudian mengakhiri klaster. Anda juga dapat menutup notebook yang melekat pada satu klaster berjalan dan beralih ke yang lain. Beberapa pengguna dapat melampirkan notebook ke klaster yang sama secara bersamaan dan berbagi file notebook di Amazon S3 dengan satu sama lain. Fitur ini memungkinkan Anda menjalankan klaster sesuai permintaan untuk menghemat biaya, dan mengurangi waktu yang dihabiskan untuk mengonfigurasi ulang notebook untuk berbagai klaster dan set data.

Anda juga dapat menjalankan notebook EMR secara terprogram menggunakan Amazon EMR API, tanpa perlu berinteraksi dengan konsol EMR Amazon (“eksekusi tanpa kepala”). Anda perlu menyertakan sel di EMR notebook yang memiliki tanda parameter. Sel tersebut memungkinkan script untuk meneruskan nilai input baru pada notebook. Notebook berparameter dapat digunakan kembali dengan set yang berbeda dari nilai input. Tidak perlu membuat salinan notebook yang sama untuk mengedit dan mengeksekusi dengan nilai input baru. Amazon EMR membuat dan menyimpan notebook keluaran pada S3 untuk setiap proses notebook berparameter. Untuk sampel kode API EMR notebook, lihat Contoh perintah untuk menjalankan EMR Notebooks secara programatis.

> **Important**
>
> Kemampuan EMR Notebooks mendukung cluster yang menggunakan Amazon EMR rilis 5.18.0 dan lebih tinggi. Kami menyarankan Anda menggunakan EMR Notebooks dengan cluster yang menggunakan Amazon EMR versi terbaru, atau setidaknya 5.30.0, 5.32.0, atau 6.2.0. Dengan rilis ini, kernel Jupyter berjalan di cluster terlampir daripada pada instance Jupyter. Ini meningkatkan kinerja dan meningkatkan kemampuan Anda untuk menyesuaikan kernel dan pustaka. Untuk informasi selengkapnya, lihat Perbedaan kemampuan dengan versi rilis klaster.

Berlaku biaya untuk penyimpanan Amazon S3 dan untuk klaster Amazon EMR.

Amazon EMR Notebooks tersedia sebagai Amazon EMR Studio Workspaces di konsol baru

Membuat transisi dari EMR Notebooks ke Workspaces

Di konsol Amazon EMR yang baru, kami telah menggabungkan EMR Notebooks dengan Amazon EMR Studio Workspaces menjadi satu pengalaman. Saat menggunakan EMR Studio, Anda dapat membuat dan mengonfigurasi Ruang Kerja yang berbeda untuk mengatur dan menjalankan buku catatan. Jika Anda memiliki notebook EMR Amazon di konsol lama, mereka tersedia sebagai EMR Studio Workspaces di konsol baru.

Amazon EMR membuat EMR Studio Workspaces baru ini untuk Anda. Jumlah Studios yang kami buat sesuai dengan jumlah VPC berbeda yang Anda gunakan dari EMR Notebooks. Misalnya, jika Anda terhubung ke cluster EMR di dua VPC berbeda dari EMR Notebooks, maka kami membuat dua EMR Studios baru. Notebook Anda didistribusikan di antara Studios baru.

> **Important**
>
> Kami mematikan opsi untuk membuat notebook baru di konsol EMR Amazon lama. Sebagai gantinya, gunakan Create Workspace di konsol Amazon EMR baru.

Untuk informasi selengkapnya tentang Amazon EMR Studio Workspaces, lihat. Pelajari dasar-dasar Ruang Kerja Untuk ikhtisar konseptual EMR Studio, Workspace lihat di halamanCara Kerja Amazon EMR Studio.

Apa yang perlu Anda lakukan?

Meskipun Anda masih dapat menggunakan notebook yang ada di konsol lama, sebaiknya gunakan Amazon EMR Studio Workspaces di konsol baru. Anda harus mengonfigurasi izin peran tambahan untuk mengaktifkan kemampuan di EMR Studio yang tidak tersedia di EMR Notebooks.

> **Note**
>
> Minimal, untuk melihat EMR Notebooks yang ada sebagai EMR Studio Workspaces dan untuk membuat Workspaces baru, pengguna `elasticmapreduce:ListStudios` harus memiliki dan izin pada peran mereka. `elasticmapreduce:CreateStudioPresignedUrl` Untuk mengakses semua fitur EMR Studio, lihat Mengaktifkan fitur EMR Studio untuk pengguna EMR Notebooks daftar lengkap izin tambahan yang dibutuhkan pengguna EMR Notebooks.

Kemampuan yang ditingkatkan di EMR Studio di luar EMR Notebooks

Dengan Amazon EMR Studio, Anda dapat mengatur dan menggunakan kemampuan berikut yang tidak tersedia dengan EMR Notebooks:

- Jelajahi dan lampirkan ke cluster EMR dari dalam Jupyterlab
- Jelajahi dan lampirkan ke cluster virtual EMR Notebooks dari dalam Jupyterlab
- Connect ke repo Git dari dalam Jupyterlab
- Berkolaborasi dengan anggota tim Anda yang lain untuk menulis dan menjalankan kode buku catatan
- Jelajahi data dengan SQL Explorer
- Penyediaan kluster EMR dengan Service Catalog

Untuk daftar lengkap kemampuan dengan Amazon EMR Studio, lihat. Fitur utama dari EMR Studio

Mengaktifkan fitur EMR Studio untuk pengguna EMR Notebooks

EMR Studios baru yang akan kami buat sebagai bagian dari penggabungan ini menggunakan peran `EMR_Notebooks_DefaultRole` IAM yang ada sebagai peran layanan EMR Studio.

Pengguna yang beralih ke EMR Studio dari EMR Notebooks dan ingin menggunakan kemampuan tambahan EMR Studio memerlukan beberapa izin peran baru. Tambahkan izin berikut ke peran pengguna EMR Notebooks Anda yang berencana menggunakan EMR Studio.

> **Note**
>
> Minimal, untuk melihat EMR Notebooks yang ada sebagai EMR Studio Workspaces dan untuk membuat Workspaces baru, pengguna `elasticmapreduce:ListStudios` harus memiliki dan izin pada peran mereka. `elasticmapreduce:CreateStudioPresignedUrl` Untuk menggunakan semua fitur EMR Studio, tambahkan semua izin yang tercantum di bawah ini. Pengguna admin juga memerlukan izin untuk membuat dan mengelola EMR Studio. Untuk informasi selengkapnya, lihat Izin administrator untuk membuat dan mengelola EMR Studio.

```
"elasticmapreduce:DescribeStudio",
"elasticmapreduce:ListStudios",
"elasticmapreduce:CreateStudioPresignedUrl",
"elasticmapreduce:UpdateEditor",
"elasticmapreduce:PutWorkspaceAccess",
"elasticmapreduce:DeleteWorkspaceAccess",
"elasticmapreduce:ListWorkspaceAccessIdentities",
"emr-containers:ListVirtualClusters",
"emr-containers:DescribeVirtualCluster",
"emr-containers:ListManagedEndpoints",
"emr-containers:DescribeManagedEndpoint",
"emr-containers:CreateAccessTokenForManagedEndpoint",
"emr-containers:ListJobRuns",
"emr-containers:DescribeJobRun",
"servicecatalog:SearchProducts",
"servicecatalog:DescribeProduct",
"servicecatalog:DescribeProductView",
"servicecatalog:DescribeProvisioningParameters",
"servicecatalog:ProvisionProduct",
"servicecatalog:UpdateProvisionedProduct",
```

```
"servicecatalog:ListProvisioningArtifacts",
"servicecatalog:DescribeRecord",
"servicecatalog:ListLaunchPaths",
"cloudformation:DescribeStackResources"
```

Izin berikut juga diperlukan untuk menggunakan kemampuan kolaborasi di EMR Studio, tetapi tidak diperlukan dengan EMR Notebooks.

```
"sso-directory:SearchUsers",
"iam:GetUser",
"iam:GetRole",
"iam:ListUsers",
"iam:ListRoles",
"sso:GetManagedApplicationInstance"
```

Pertimbangan saat menggunakan EMR Notebooks

> **Note**
>
> EMR Notebooks tersedia sebagai EMR Studio Workspaces di konsol baru. Anda masih dapat menggunakan notebook yang ada di konsol lama, tetapi Anda tidak dapat membuat notebook baru di konsol lama. Tombol Create Workspace di konsol baru menggantikan fungsi ini. Untuk mengakses atau membuat Ruang Kerja, pengguna EMR Notebooks memerlukan izin peran IAM tambahan. Untuk informasi selengkapnya, lihat Amazon EMR Notebook adalah Amazon EMR Studio Workspaces di konsol baru dan Apa yang baru di konsol?

Pertimbangkan persyaratan berikut ketika Anda membuat klaster dan mengembangkan solusi menggunakan EMR notebook.

Persyaratan klaster

- Aktifkan Amazon EMR Block Public Access — Akses masuk ke klaster memungkinkan pengguna klaster untuk mengeksekusi kernel notebook. Pastikan bahwa hanya pengguna yang diotorisasi yang dapat mengakses klaster. Kami sangat menyarankan Anda membiarkan block public access diaktifkan, dan Anda membatasi lalu lintas SSH masuk hanya untuk sumber tepercaya. Untuk informasi lebih lanjut, lihat Menggunakan Akses publik blok Amazon EMR dan Mengendalikan lalu lintas jaringan dengan grup keamanan.

- Menggunakan Klaster Kompatibel — Sebuah klaster yang terpasang pada notebook harus memenuhi persyaratan berikut:
  - Hanya klaster yang dibuat menggunakan Amazon EMR yang didukung. Anda dapat membuat sebuah klaster secara independen dalam Amazon EMR dan kemudian melampirkan EMR notebook, atau Anda dapat membuat klaster kompatibel ketika Anda membuat EMR notebook.
  - Hanya klaster yang dibuat menggunakan rilis Amazon EMR versi 5.18.0 dan yang lebih baru yang didukung. Lihat the section called “Perbedaan kemampuan dengan versi rilis klaster”.
  - Klaster yang dibuat menggunakan instans Amazon EC2 dengan prosesor AMD EPYC—misalnya, contoh instans m5a.* dan r5a.*—tidak didukung.
  - EMR Notebooks hanya berfungsi dengan klaster yang dibuat dengan `VisibleToAllUsers` diatur ke `true`. `VisibleToAllUsers` adalah `true` secara default.
  - Klaster harus diluncurkan dalam EC2-VPC. Subnet publik dan privat didukung. Platform EC2 Klasik tidak didukung.
  - Klaster harus diluncurkan dengan Hadoop, Spark, dan Livy yang diinstal. Aplikasi lain dapat diinstal, tetapi EMR Notebooks saat ini hanya mendukung klaster Spark.

    > **Important**
    >
    > Untuk versi rilis Amazon EMR 5.32.0 dan yang lebih baru, atau 6.2.0 dan yang lebih baru, cluster Anda juga harus menjalankan aplikasi Jupyter Enterprise Gateway agar dapat bekerja dengan EMR Notebooks.

  - Klaster yang menggunakan autentikasi Kerberos tidak didukung.
  - Klaster terintegrasi dengan AWS Lake Formation mendukung pemasangan pustaka notebook saja. Menginstal kernel dan pustaka di klaster tidak didukung.
  - Cluster dengan beberapa node primer tidak didukung.
  - Klaster menggunakan instans Amazon EC2 berdasarkan AWS Graviton2 tidak didukung.

Perbedaan kemampuan dengan versi rilis klaster

Kami sangat menyarankan agar Anda menggunakan EMR Notebooks dengan klaster yang dibuat menggunakan Amazon EMR versi rilis 5.30.0, 5.32.0, atau lebih baru, atau 6.2.0 atau lebih baru. Dengan versi ini, EMR Notebooks menjalankan kernel pada klaster Amazon EMR yang dilampirkan. Kernel dan pustaka dapat diinstal langsung pada node primer cluster. Menggunakan EMR Notebooks dengan versi klaster ini memiliki manfaat sebagai berikut:

- Peningkatan kinerja — Kernel Notebook berjalan pada klaster dengan jenis instans EC2 yang Anda pilih. Versi sebelumnya menjalankan kernel pada instans khusus yang tidak dapat diubah ukurannya, diakses, atau disesuaikan.
- Kemampuan untuk menambah dan menyesuaikan kernel — Anda dapat terhubung ke klaster untuk menginstal paket kernel menggunakan `conda` dan `pip`. Selain itu, instalasi `pip` didukung menggunakan perintah terminal dalam sel notebook. Di versi sebelumnya, hanya kernel pra-instal yang tersedia (Python,, Spark PySpark, dan SparkR). Untuk informasi selengkapnya, lihat Menginstal kernel dan pustaka Python pada node primer cluster.
- Kemampuan untuk menginstal pustaka Python — Anda dapat menginstal pustaka Python pada node utama cluster menggunakan dan. `conda` `pip` Kami merekomendasikan penggunaan `conda`. Dengan versi sebelumnya, hanya library dengan cakupan notebook untuk yang didukung. PySpark

Fitur EMR Notebooks yang didukung oleh rilis klaster

Versi rilis klaster	Pustaka dengan cakupan notebook untuk PySpark	Instalasi kernel di klaster	Instalasi pustaka Python pada simpul utama
Lebih awal dari 5.18.0	EMR Notebooks tidak didukung		
5.18.0–5.25.0	Tidak	Tidak	Tidak
5.26.0-5.29.0	Ya	Tidak	Tidak
5.30.0	Ya	Ya	Ya
6.0.0	Tidak	Tidak	Tidak
5.32.0 dan yang lebih baru, dan 6.2.0 dan yang lebih baru	Ya	Ya	Ya

Batas untuk EMR Notebooks yang terpasang bersamaan

Saat Anda membuat klaster yang mendukung notebook, pertimbangkan jenis Instans EC2 dari simpul utama klaster. Batasan memori instans EC2 ini menentukan jumlah notebook yang dapat siap secara bersamaan untuk menjalankan kode dan kueri pada klaster.

Tipe instans EC2 simpul primer	Jumlah EMR Notebooks
*.medium	2
*.large	4
*.xlarge	8
*.2xlarge	16
*.4xlarge	24
*.8xlarge	24
*.16xlarge	24

Versi Jupyter Notebook dan Python

EMR Notebooks menjalankan Jupyter Notebook versi 6.0.2 dan Python 3.6.5 terlepas dari versi rilis Amazon EMR dari klaster yang menempel.

Pertimbangan terkait keamanan

Menggunakan lokasi S3 terenkripsi

Jika Anda menentukan lokasi terenkripsi di Amazon S3 untuk menyimpan file notebook, Anda harus mengatur Peran layanan untuk EMR Notebooks sebagai pengguna kunci. Peran layanan default adalah `EMR_Notebooks_DefaultRole`. Jika Anda menggunakan kunci AWS KMS untuk enkripsi, lihat Menggunakan kebijakan kunci di KMS AWS di AWS Key Management Service Panduan Developer dan artikel dukungan untuk menambahkan pengguna kunci.

Menggunakan cookie dengan domain hosting

Untuk meningkatkan keamanan aplikasi off-console yang mungkin Anda gunakan dengan Amazon EMR, domain hosting aplikasi terdaftar di Daftar Akhiran Publik (PSL). Contoh domain hosting ini meliputi:`emrstudio-prod.us-east-1.amazonaws.com`,`emrnotebooks-prod.us-east-1.amazonaws.com`,`emrappui-prod.us-east-1.amazonaws.com`. Untuk keamanan lebih lanjut, jika Anda perlu mengatur cookie sensitif di nama domain default, kami sarankan Anda menggunakan cookie dengan `__Host-` awalan. Ini membantu

mempertahankan domain Anda dari upaya pemalsuan permintaan lintas situs (CSRF). Untuk informasi selengkapnya, lihat Set-Cookiehalaman di Jaringan Pengembang Mozilla.

Membuat Notebook

> **Note**
>
> EMR Notebooks tersedia sebagai EMR Studio Workspaces di konsol baru. Anda masih dapat menggunakan notebook yang ada di konsol lama, tetapi Anda tidak dapat membuat notebook baru di konsol lama. Tombol Create Workspace di konsol baru menggantikan fungsi ini. Untuk mengakses atau membuat Ruang Kerja, pengguna EMR Notebooks memerlukan izin peran IAM tambahan. Untuk informasi selengkapnya, lihat Amazon EMR Notebook adalah Amazon EMR Studio Workspaces di konsol baru dan Apa yang baru di konsol?

Anda membuat notebook EMR menggunakan konsol EMR Amazon lama. Membuat notebook menggunakan AWS CLI atau API Amazon EMR tidak didukung.

Untuk membuat EMR notebook

1. Buka konsol Amazon EMR di https://console.aws.amazon.com/elasticmapreduce/.
2. Pilih Notebook, Buat notebook.
3. Masukkan Nama notebook dan Deskripsi notebook opsional.
4. Jika Anda memiliki klaster aktif yang Anda ingin tempelkan dengan notebook, biarkan default Pilih klaster yang ada dipilih, klik Pilih, pilih sebuah klaster dari daftar, dan kemudian klik Pilih klaster. Untuk informasi tentang persyaratan klaster untuk EMR Notebooks, lihat Pertimbangan saat menggunakan EMR Notebooks.

   —atau—

   Pilih Buat klaster, masukkan Nama klaster dan pilih opsi sesuai dengan pedoman berikut. Klaster dibuat di VPC default untuk akun yang menggunakan instans Sesuai permintaan.

Pengaturan	Deskripsi
Nama cluster	Nama familier yang digunakan untuk mengidentifikasi klaster.

Pengaturan	Deskripsi
Rilis	Tidak dapat diubah. Default ke versi rilis Amazon EMR terbaru (5.36.1).
Aplikasi	Tidak dapat diubah. Daftar aplikasi yang diinstal pada klaster.
Contoh	Masukkan jumlah instans dan pilih jenis Instans EC2. Satu contoh digunakan untuk node primer. Sisanya digunakan untuk simpul inti. Jenis instans menentukan jumlah notebook yang dapat ditempelk an ke klaster secara bersamaan. Untuk informasi selengkapnya, lihat Batas untuk EMR Notebooks yang terpasang bersamaan.
Peran EMR	Biarkan default atau pilih tautan untuk menentukan peran layanan kustom untuk Amazon EMR. Untuk informasi selengkap nya, lihat Peran layanan untuk Amazon EMR (peran EMR).
Profil instans EC2	Biarkan default atau pilih tautan untuk menentukan peran layanan kustom untuk instans EC2. Untuk informasi selengkapnya, lihat Peran layanan untuk instans EC2 klaster (profil instans EC2).
EC2 key pair	Pilih pasangan kunci EC2 untuk dapat terhubung ke instans klaster. Untuk informasi selengkapnya, lihat Connect ke node utama menggunakan SSH.

Pengaturan	Deskripsi
Pengakhiran otomatis	Pengakhiran otomatis didukung untuk Amazon EMR versi 5.30.0 dan 6.1.0 dan yang lebih baru. Pilih kotak centang untuk mengaktifkan penghentian otomatis, lalu tentukan jumlah waktu idle setelah cluster akan mati secara otomatis. Untuk informasi selengkapnya, lihat Menggunakan kebijakan penghentian otomatis.

5. Untuk Grup keamanan, pilih Gunakan grup keamanan default. Atau, pilih Pilih grup keamanan dan pilih grup keamanan kustom yang tersedia di VPC klaster. Anda memilih satu untuk instance utama dan satu lagi untuk instance klien notebook. Untuk informasi selengkapnya, lihat the section called “Grup keamanan untuk EMR Notebooks”.
6. Untuk AWS Peran Layanan, biarkan default atau pilih peran kustom dari daftar. Instans klien untuk notebook menggunakan peran ini. Untuk informasi selengkapnya, lihat Peran layanan untuk EMR Notebooks.
7. Untuk Lokasi notebook Pilih lokasi di Amazon S3 tempat file notebook disimpan, atau tentukan lokasi Anda sendiri. Jika bucket dan folder tidak ada, Amazon EMR membuatnya.

   Amazon EMR membuat folder dengan ID Notebook sebagai nama folder, dan menyimpan notebook ke file bernama *`NotebookName`*`.ipynb`. Misalnya, jika Anda menentukan lokasi Amazon S3 `s3://MyBucket/MyNotebooks` untuk notebook bernama `MyFirstEMRManagedNotebook`, file notebook disimpan ke `s3://MyBucket/MyNotebooks/`*`NotebookID`*`/MyFirstEMRManagedNotebook.ipynb`.

   Jika Anda menentukan lokasi terenkripsi di Amazon S3, Anda harus mengatur Peran layanan untuk EMR Notebooks sebagai pengguna kunci. Peran layanan default adalah `EMR_Notebooks_DefaultRole`. Jika Anda menggunakan kunci AWS KMS untuk enkripsi, lihat Menggunakan kebijakan kunci di KMS AWS di AWS Key Management Service Panduan Developer dan artikel dukungan untuk menambahkan pengguna kunci.
8. Atau, jika Anda telah menambahkan repositori berbasis Git ke Amazon EMR yang ingin Anda kaitkan dengan notebook ini, pilih Repositori Git, pilih Pilih repositori lalu pilih repositori dari

daftar. Untuk informasi selengkapnya, lihat Mengasosiasikan repositori berbasis Git dengan EMR Notebooks.

9. Atau, pilih Tanda, dan kemudian tambahkan tanda kunci-nilai tambahan untuk notebook.

> **Important**
>
> Sebuah tanda default dengan set string Kunci diatur ke `creatorUserID` dan set nilai yang ditetapkan ke ID pengguna IAM Anda diterapkan untuk tujuan akses. Kami menyarankan agar Anda tidak mengubah atau menghapus tanda ini karena dapat digunakan untuk mengontrol akses. Untuk informasi selengkapnya, lihat Gunakan klaster dan tanda Notebook dengan kebijakan IAM untuk kendali akses.

10. Pilih Buat Notebook.

Bekerja dengan EMR Notebooks

> **Note**
>
> EMR Notebooks tersedia sebagai EMR Studio Workspaces di konsol baru. Anda masih dapat menggunakan notebook yang ada di konsol lama, tetapi Anda tidak dapat membuat notebook baru di konsol lama. Tombol Create Workspace di konsol baru menggantikan fungsi ini. Untuk mengakses atau membuat Ruang Kerja, pengguna EMR Notebooks memerlukan izin peran IAM tambahan. Untuk informasi selengkapnya, lihat Amazon EMR Notebook adalah Amazon EMR Studio Workspaces di konsol baru dan Apa yang baru di konsol?

Setelah Anda membuat notebook EMR, notebook membutuhkan waktu singkat untuk memulai. Status di daftar Notebook menunjukkan Memulai. Anda bisa membuka notebook saat statusnya Siap. Mungkin butuh waktu sedikit lebih lama untuk notebook menjadi Siap jika Anda membuat sebuah klaster bersama dengannya.

> **Tip**
>
> Refresh browser Anda atau pilih ikon refresh di atas daftar notebook untuk menyegarkan status notebook.

Memahami status Notebook

EMR notebook dapat memiliki hal berikut untuk Status di daftar Notebook.

Status	Arti
Siap	Anda bisa membuka notebook menggunakan editor notebook. Sementara notebook memiliki status Siap, Anda dapat menghentikan atau menghapusnya. Untuk mengganti klaster, Anda harus menghentikan notebook terlebih dahulu. Jika notebook di status Siap idle untuk jangka waktu yang lama, notebook dihentikan secara otomatis.
Mulai	Notebook sedang dibuat dan ditempelkan ke klaster. Saat notebook dimulai, Anda tidak dapat membuka editor notebook, menghenti kannya, menghapusnya, atau mengubah klaster.
Tertunda	Notebook telah dibuat, dan sedang menunggu integrasi dengan klaster selesai. Klaster mungkin masih menyediakan sumber daya atau menanggapi permintaan lainnya. Anda bisa membuka editor notebook dengan notebook dalam mode lokal. Kode apa pun yang bergantung pada proses klaster tidak mengeksekusi dan gagal.
Berhenti	Notebook dimatikan, atau klaster yang ditempelkan pada notebook berakhir. Saat notebook berhenti, Anda tidak dapat membuka editor notebook, menghentikannya, menghapus nya, atau mengubah klaster.
Dihentikan	Notebook telah dimatikan. Anda dapat memulai notebook pada klaster yang sama, selama

Status	Arti
	klaster masih berjalan. Anda dapat mengubah klaster, dan menghapus klaster.
Menghapus	Klaster sedang dihapus dari daftar klaster yang tersedia. File notebook, *NotebookName* `.ipynb` tetap di Amazon S3 dan terus menambah biaya penyimpanan yang berlaku.

Bekerja dengan editor Notebook

Keuntungan menggunakan notebook EMR adalah Anda dapat meluncurkan notebook di Jupyter atau JupyterLab langsung dari konsol.

Dengan EMR Notebooks, editor notebook yang Anda akses dari konsol Amazon EMR adalah editor Notebook Jupyter sumber terbuka yang sudah dikenal atau. JupyterLab Karena editor notebook diluncurkan dalam konsol Amazon EMR, lebih efisien untuk mengonfigurasi akses daripada dengan notebook yang di-host pada klaster Amazon EMR. Anda tidak perlu mengonfigurasi klien pengguna untuk membuat akses web melewati SSH, aturan grup keamanan, dan konfigurasi proxy. Jika pengguna memiliki izin yang memadai, mereka hanya dapat membuka editor notebook dalam konsol Amazon EMR.

Hanya satu pengguna dapat memiliki EMR notebook terbuka pada satu waktu dari dalam Amazon EMR. Jika pengguna lain mencoba membuka EMR notebook yang sudah terbuka, terjadi kesalahan.

> **Important**
>
> Amazon EMR menciptakan URL pre-signed unik untuk setiap sesi editor notebook, yang hanya berlaku untuk waktu yang singkat. Kami menyarankan agar Anda tidak membagikan URL editor notebook. Melakukan hal ini akan menimbulkan risiko keamanan karena penerima URL mengadopsi izin Anda untuk mengedit notebook dan menjalankan kode notebook selama masa hidup URL. Jika orang lain memerlukan akses ke buku catatan, berikan izin kepada pengguna mereka melalui kebijakan izin dan pastikan bahwa peran layanan untuk EMR Notebooks memiliki akses ke lokasi Amazon S3. Untuk informasi lebih lanjut, lihat the section called “Keamanan” dan Peran layanan untuk EMR Notebooks.

Untuk membuka editor notebook untuk EMR notebook

1. Pilih notebook dengan Status dari Siap atau Tertunda dari daftar Notebook.
2. Pilih Buka di JupyterLab atau Buka di Jupyter.

   Tab browser baru terbuka ke editor JupyterLab atau Jupyter Notebook.
3. Dari menu Kernel, pilih Ubah kernel lalu pilih kernel untuk bahasa pemrograman Anda.

   Anda sekarang siap untuk menulis dan menjalankan kode dari dalam editor notebook.

Menyimpan isi Notebook

Ketika Anda bekerja di editor notebook, isi sel notebook dan output disimpan secara otomatis ke file notebook secara berkala di Amazon S3. Notebook yang tidak memiliki perubahan sejak terakhir kali sel diedit menunjukkan (disimpan otomatis) di samping nama notebook di editor. Jika perubahan belum disimpan, perubahan belum disimpan muncul.

Anda bisa menyimpan notebook secara manual. Dari menu File, pilih Simpan dan Checkpoint atau tekan CTRL+S. Ini membuat file bernama *NotebookName*`.ipynb` dalam folder checkpoint dalam folder notebook di Amazon S3. Sebagai contoh, `s3://`*MyBucket*`/`*MyNotebookFolder*`/`*NotebookID*`/checkpoints/`*NotebookName*`.ipynb`. Hanya file checkpoint terbaru yang disimpan di lokasi ini.

Mengubah klaster

Anda dapat mengubah klaster yang ditempelkan EMR notebook tanpa mengubah isi notebook itu sendiri. Anda dapat mengubah klaster hanya untuk mereka notebook yang memiliki status Dihentikan.

Untuk mengubah klaster EMR notebook

1. Jika notebook yang ingin Anda ubah sedang berjalan, pilih dari daftar Notebook dan pilih Berhenti.
2. Ketika status notebook Dihentikan, pilih notebook dari daftar Notebook, dan kemudian pilih Tampilkan detail.
3. Pilih Ubah klaster.

4. Jika Anda memiliki klaster aktif yang menjalankan Hadoop, Spark, dan Livy yang Anda ingin tempelkan pada notebook, biarkan default, dan pilih klaster dari daftar. Hanya klaster yang memenuhi persyaratan terdaftar.

   — atau —

   Pilih Buat klaster lalu pilih opsi klaster. Untuk informasi selengkapnya, lihat Persyaratan klaster.

5. Pilih satu opsi untuk Grup keamanan, lalu pilih Ubah klaster dan mulai notebook.

Menghapus Notebook dan file Notebook

Saat Anda menghapus EMR notebook menggunakan konsol Amazon EMR, Anda menghapus notebook dari daftar notebook yang tersedia. Namun, file notebook, tetap di Amazon S3 dan terus menambah biaya penyimpanan yang berlaku.

Untuk menghapus notebook dan menghapus file terkait

1. Buka konsol Amazon EMR di https://console.aws.amazon.com/elasticmapreduce/.
2. Pilih Notebook, pilih notebook Anda dari daftar, lalu pilih Tampilkan detail.
3. Pilih ikon folder di sebelah Lokasi notebook dan salin URL, yang ada dalam pola `s3://MyNotebookLocationPath/NotebookID/`.
4. Pilih Hapus.

   Notebook dihapus dari daftar, dan detail notebook tidak dapat lagi dilihat.

5. Ikuti petunjuk untuk Bagaimana cara menghapus folder dari bucket S3? di Panduan Pengguna Layanan Penyimpanan Sederhana Amazon. Arahkan ke bucket dan folder dari langkah 3.

   — atau —

   Jika Anda memiliki AWS CLI diinstal, buka prompt perintah dan ketik perintah di akhir paragraf ini. Ganti lokasi Amazon S3 dengan lokasi yang Anda salin di atas. Pastikan bahwa AWS CLI dikonfigurasi dengan access key pengguna dengan izin untuk menghapus lokasi Amazon S3. Untuk informasi lebih lanjut, lihat Mengonfigurasi AWS CLI di AWS Command Line Interface Panduan Pengguna.

   ```
   aws s3 rm s3://MyNotebookLocationPath/NotebookID
   ```

Berbagi file Notebook

Setiap EMR notebook disimpan ke Amazon S3 sebagai file bernama *NotebookName*.ipynb. Selama file notebook kompatibel dengan versi yang sama dari Jupyter Notebook yang didasarkan pada EMR Notebooks, Anda dapat membuka notebook sebagai EMR notebook.

Cara termudah untuk membuka file notebook dari pengguna lain adalah dengan menyimpan file*.ipynb dari pengguna lain ke sistem file lokal Anda, lalu gunakan fitur unggah di Jupyter dan editor. JupyterLab

Anda dapat menggunakan proses ini untuk menggunakan EMR notebook yang dibagikan oleh orang lain, notebook yang dibagikan di komunitas Jupyter, atau untuk memulihkan notebook yang telah dihapus dari konsol saat Anda masih memiliki file notebook.

Untuk menggunakan file notebook yang berbeda sebagai dasar untuk EMR notebook

1. Sebelum melanjutkan, tutup editor notebook untuk notebook apa pun yang akan Anda gunakan, lalu hentikan notebook jika itu adalah EMR notebook.
2. Buat EMR notebook dan masukkan nama untuknya. Nama yang Anda masukkan untuk notebook akan menjadi nama file yang perlu Anda ganti. Nama file baru harus cocok dengan nama file ini persis.
3. Buat catatan dari lokasi di Amazon S3 yang Anda pilih untuk notebook. File yang Anda ganti dalam folder dengan jejak dan nama file seperti pola berikut: s3://*MyNotebookLocation*/*NotebookID*/*MyNotebookName*.ipynb.
4. Hentikan notebook.
5. Ganti file notebook lama di lokasi Amazon S3 dengan yang baru, dengan menggunakan nama yang persis sama.

   AWS CLIPerintah berikut untuk Amazon S3 menggantikan file yang disimpan ke mesin lokal yang disebut notebook SharedNotebook.ipynb EMR dengan nama MyNotebook, ID darie-12A3BCDEFJHIJKLMNO45PQRST, dan dibuat dengan ditentukan di MyBucket/MyNotebooksFolder Amazon S3. Untuk informasi tentang menggunakan konsol Amazon S3 untuk menyalin dan mengganti file, lihat Mengunggah, mengunduh, dan mengelola objek di Panduan Pengguna Layanan Penyimpanan Sederhana Amazon.

   ```
   aws s3 cp SharedNotebook.ipynb s3://MyBucket/
   MyNotebooksFolder/-12A3BCDEFJHIJKLMNO45PQRST/MyNotebook.ipynb
   ```

Contoh perintah untuk menjalankan EMR Notebooks secara programatis

> **Note**
>
> EMR Notebooks tersedia sebagai EMR Studio Workspaces di konsol baru. Anda masih dapat menggunakan notebook yang ada di konsol lama, tetapi Anda tidak dapat membuat notebook baru di konsol lama. Tombol Create Workspace di konsol baru menggantikan fungsi ini. Untuk mengakses atau membuat Ruang Kerja, pengguna EMR Notebooks memerlukan izin peran IAM tambahan. Untuk informasi selengkapnya, lihat Amazon EMR Notebook adalah Amazon EMR Studio Workspaces di konsol baru dan Apa yang baru di konsol?

Gambaran Umum

Anda dapat menjalankan notebook EMR dengan API eksekusi dari skrip atau dari baris perintah. Saat Anda memulai, menghentikan, membuat daftar, dan menjelaskan eksekusi notebook EMR di luar AWS konsol, Anda dapat mengontrol notebook EMR secara terprogram. Anda dapat meneruskan nilai parameter yang berbeda ke buku catatan dengan sel notebook berparameter. Ini menghilangkan kebutuhan untuk membuat salinan notebook untuk setiap set nilai parameter baru. Untuk informasi selengkapnya, lihat tindakan Amazon EMR API.

Anda dapat menjadwalkan atau mengelompokkan eksekusi notebook EMR dengan acara Amazon CloudWatch dan. AWS Lambda Untuk informasi selengkapnya, lihat Menggunakan AWS Lambda dengan CloudWatch Acara Amazon.

Izin peran untuk eksekusi terprogram

Untuk menggunakan eksekusi terprogram dengan EMR Notebooks, Anda harus mengonfigurasi izin pengguna dengan kebijakan berikut:

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

```
                "elasticmapreduce:StartNotebookExecution",
                "elasticmapreduce:DescribeNotebookExecution",
                "elasticmapreduce:ListNotebookExecutions"
            ],
            "Resource": "*"
        },
        {
            "Sid": "AllowPassingServiceRole",
            "Effect": "Allow",
            "Action": [
                "iam:PassRole"
            ],
            "Resource": "arn:aws:iam::account-id:role/EMR_Notebooks_DefaultRole"
        }
    ]
}
```

Saat menjalankan EMR Notebooks secara terprogram di kluster EMR Notebooks, Anda harus menambahkan izin tambahan ini:

```
{
    "Version": "2012-10-17",
    "Statement": [
        {
            "Sid": "AllowRetrievingManagedEndpointCredentials",
            "Effect": "Allow",
            "Action": [
                "emr-containers:GetManagedEndpointSessionCredentials"
            ],
            "Resource": [
                "arn:aws:emr-containers:region:account-id:/virtualclusters/virtual-
cluster-id/endpoints/managed-endpoint-id"
            ],
            "Condition": {
                "StringEquals": {
                    "emr-containers:ExecutionRoleArn": [
                        "arn:aws:iam::account-id:role/emr-on-eks-execution-role"
                    ]
                }
            }
        },
        {
            "Sid": "AllowDescribingManagedEndpoint",
```

```
            "Effect": "Allow",
            "Action": [
                "emr-containers:DescribeManagedEndpoint"
            ],
            "Resource": [
                "arn:aws:emr-containers:region:account-id:/virtualclusters/virtual-
cluster-id/endpoints/managed-endpoint-id"
            ]
        }
    ]
}
```

Keterbatasan dengan eksekusi terprogram

- Maksimal 100 eksekusi bersamaan didukung Wilayah AWS per akun.
- Eksekusi dihentikan jika berjalan selama lebih dari 30 hari.
- Eksekusi terprogram notebook tidak didukung dengan aplikasi interaktif Amazon EMR Serverless.

Contoh eksekusi notebook EMR terprogram

Bagian berikut memberikan beberapa contoh eksekusi notebook EMR terprogram denganAWS CLI, Boto3 SDK (Python), dan Ruby:

- Contoh perintah CLI eksekusi notebook
- Sampel eksekusi notebook Python
- Sampel eksekusi notebook Ruby

Anda juga dapat menjalankan notebook berparameter sebagai bagian dari alur kerja terjadwal dengan alat orkestrasi seperti Apache Airflow atau Amazon Managed Workflows for Apache Airflow (MWAA). Untuk informasi selengkapnya, lihat Mengatur pekerjaan analitik di EMR Notebooks menggunakan MWAA di Big Data Blog. AWS

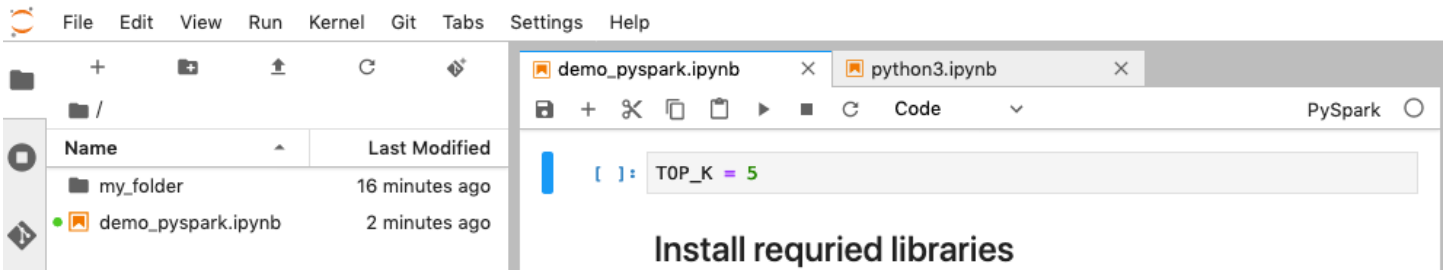
Contoh perintah CLI eksekusi notebook

> **Note**
>
> EMR Notebooks tersedia sebagai EMR Studio Workspaces di konsol baru. Anda masih dapat menggunakan notebook yang ada di konsol lama, tetapi Anda tidak dapat membuat notebook

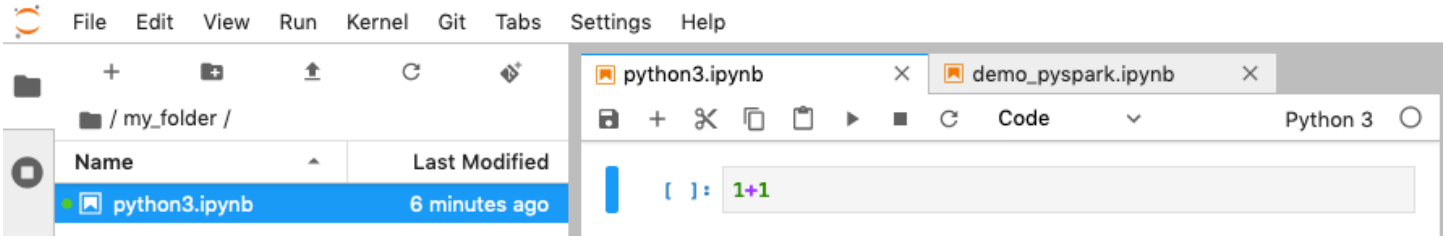
baru di konsol lama. Tombol Create Workspace di konsol baru menggantikan fungsi ini. Untuk mengakses atau membuat Ruang Kerja, pengguna EMR Notebooks memerlukan izin peran IAM tambahan. Untuk informasi selengkapnya, lihat Amazon EMR Notebook adalah Amazon EMR Studio Workspaces di konsol baru dan Apa yang baru di konsol?

Contoh berikut menggunakan notebook demo dari konsol EMR Notebooks. Untuk menemukan buku catatan, gunakan jalur file relatif ke direktori home. Dalam contoh ini, ada dua file notebook yang dapat Anda jalankan: `demo_pyspark.ipynb` dan`my_folder/python3.ipynb`.

Jalur relatif untuk file `demo_pyspark.ipynb` adalah`demo_pyspark.ipynb`, ditunjukkan di bawah ini.



Jalur relatif untuk `python3.ipynb` adalah`my_folder/python3.ipynb`, ditunjukkan di bawah ini.



Untuk informasi tentang tindakan Amazon EMR API, lihat `NotebookExecution` tindakan Amazon EMR API. .

Jalankan buku catatan

Anda dapat menggunakan AWS CLI untuk menjalankan notebook Anda dengan `start-notebook-execution` tindakan, seperti contoh berikut menunjukkan.

Example — Menjalankan notebook EMR di EMR Studio Workspace dengan klaster Amazon EMR (berjalan di Amazon EC2)

```
aws emr --region us-east-1 \
```

```
start-notebook-execution \
--editor-id e-ABCDEFG123456 \
--notebook-params '{"input_param":"my-value", "good_superhero":["superman", "batman"]}'
 \
--relative-path test.ipynb \
--notebook-execution-name my-execution \
--execution-engine '{"Id" : "j-1234ABCD123"}' \
--service-role EMR_Notebooks_DefaultRole

{
    "NotebookExecutionId": "ex-ABCDEFGHIJ1234ABCD"
}
```

Example - Menjalankan notebook EMR di Ruang Kerja EMR Studio dengan cluster EMR Notebooks

```
aws emr start-notebook-execution \
    --region us-east-1 \
    --service-role EMR_Notebooks_DefaultRole \
    --environment-variables '{"KERNEL_EXTRA_SPARK_OPTS": "--conf
 spark.executor.instances=1", "KERNEL_LAUNCH_TIMEOUT": "350"}' \
    --output-notebook-format HTML \
    --execution-engine Id=arn:aws:emr-containers:us-west-2:account-id:/
virtualclusters/ABCDEFG/
endpoints/ABCDEF,Type=EMR_ON_EKS,ExecutionRoleArn=arn:aws:iam::account-
id:role/execution-role \
    --editor-id e-ABCDEFG \
    --relative-path EMRonEKS-spark_python.ipynb
```

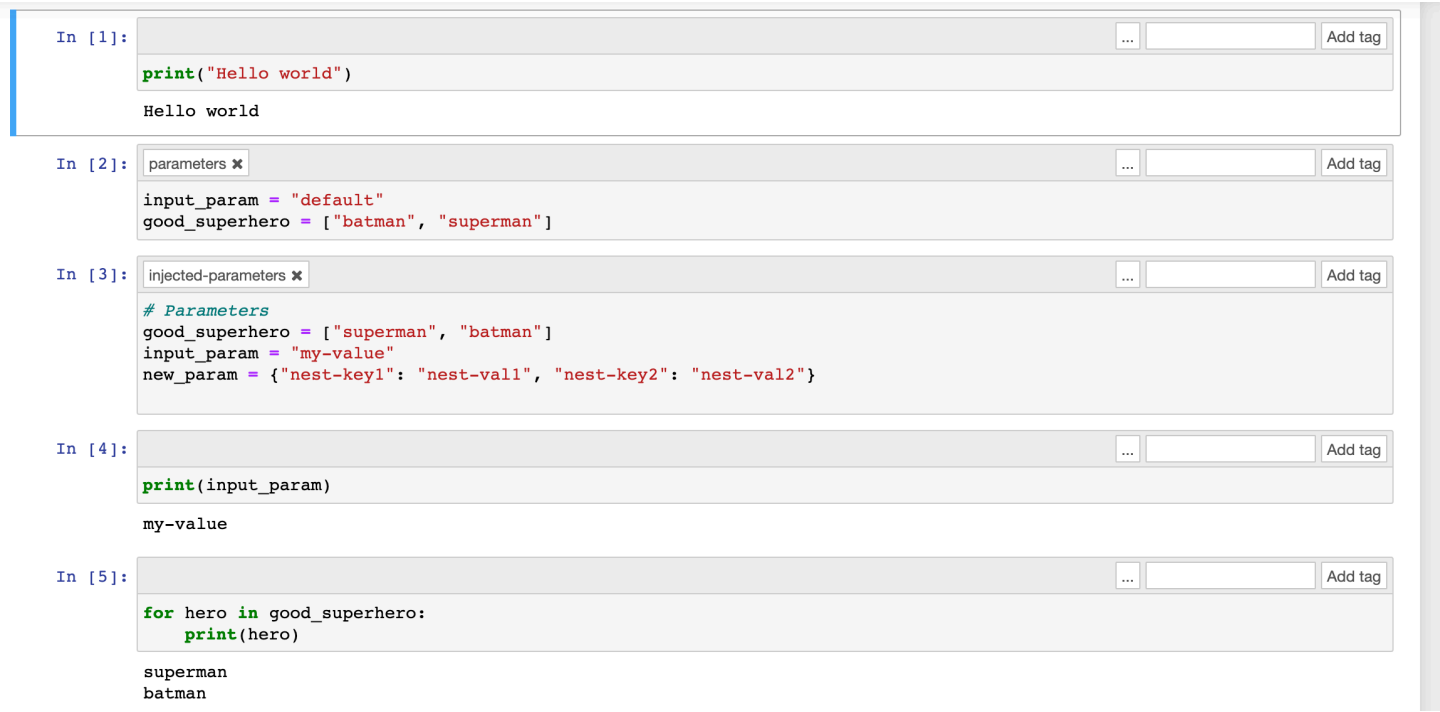
Example - Menjalankan notebook EMR yang menentukan lokasi Amazon S3-nya

```
aws emr start-notebook-execution \
    --region us-east-1 \
    --notebook-execution-name my-execution-on-emr-on-eks-cluster \
    --service-role EMR_Notebooks_DefaultRole \
    --environment-variables '{"KERNEL_EXTRA_SPARK_OPTS": "--conf
 spark.executor.instances=1", "KERNEL_LAUNCH_TIMEOUT": "350"}' \
    --output-notebook-format HTML \
    --execution-engine Id=arn:aws:emr-containers:us-west-2:account-id:/
virtualclusters/ABCDEF/
endpoints/ABCDEF,Type=EMR_ON_EKS,ExecutionRoleArn=arn:aws:iam::account-
id:role/execution-role \
    --notebook-s3-location '{"Bucket": "your-s3-bucket","Key": "s3-prefix-to-notebook-
location/EMRonEKS-spark_python.ipynb"}' \
```

```
    --output-notebook-s3-location '{"Bucket": "your-s3-bucket","Key": "s3-prefix-for-storing-output-notebook"}'
```

Keluaran notebook

Berikut adalah output dari notebook sampel. Sel 3 menunjukkan nilai parameter yang baru disuntikkan.



Jelaskan buku catatan

Anda dapat menggunakan `describe-notebook-execution` tindakan untuk mengakses informasi tentang eksekusi notebook tertentu.

```
aws emr --region us-east-1 \
describe-notebook-execution --notebook-execution-id ex-IZWZZVR9DKQ9WQ7VZWXJZR29UGHTE

{
    "NotebookExecution": {
        "NotebookExecutionId": "ex-IZWZZVR9DKQ9WQ7VZWXJZR29UGHTE",
        "EditorId": "e-BKTM2DIHXBEDRU44ANWRKIU8N",
        "ExecutionEngine": {
            "Id": "j-2QMOV6JAX1TS2",
            "Type": "EMR",
            "MasterInstanceSecurityGroupId": "sg-05ce12e58cd4f715e"
```

```
        },
        "NotebookExecutionName": "my-execution",
        "NotebookParams": "{\"input_param\":\"my-value\", \"good_superhero\":
[\"superman\", \"batman\"]}",
        "Status": "FINISHED",
        "StartTime": 1593490857.009,
        "Arn": "arn:aws:elasticmapreduce:us-east-1:123456789012:notebook-execution/ex-
IZWZZVR9DKQ9WQ7VZWXJZR29UGHTE",
        "LastStateChangeReason": "Execution is finished for cluster j-2QMOV6JAX1TS2.",
        "NotebookInstanceSecurityGroupId": "sg-0683b0a39966d4a6a",
        "Tags": []
    }
}
```

Hentikan buku catatan

Jika notebook Anda menjalankan eksekusi yang ingin Anda hentikan, Anda dapat melakukannya dengan `stop-notebook-execution` perintah.

```
# stop a running execution
aws emr --region us-east-1 \
stop-notebook-execution --notebook-execution-id ex-IZWZX78UVPAATC8LHJR129B1RBN4T


# describe it
aws emr --region us-east-1 \
describe-notebook-execution --notebook-execution-id ex-IZWZX78UVPAATC8LHJR129B1RBN4T

{
    "NotebookExecution": {
        "NotebookExecutionId": "ex-IZWZX78UVPAATC8LHJR129B1RBN4T",
        "EditorId": "e-BKTM2DIHXBEDRU44ANWRKIU8N",
        "ExecutionEngine": {
            "Id": "j-2QMOV6JAX1TS2",
            "Type": "EMR"
        },
        "NotebookExecutionName": "my-execution",
        "NotebookParams": "{\"input_param\":\"my-value\", \"good_superhero\":
[\"superman\", \"batman\"]}",
        "Status": "STOPPED",
        "StartTime": 1593490876.241,
        "Arn": "arn:aws:elasticmapreduce:us-east-1:123456789012:editor-execution/ex-
IZWZX78UVPAATC8LHJR129B1RBN4T",
```

```
        "LastStateChangeReason": "Execution is stopped for cluster j-2QMOV6JAX1TS2.
 Internal error",
        "Tags": []
    }
}
```

Buat daftar eksekusi untuk buku catatan berdasarkan waktu mulai

Anda dapat meneruskan `--from` parameter `list-notebook-executions` ke daftar eksekusi buku catatan Anda berdasarkan waktu mulai.

```
# filter by start time
aws emr --region us-east-1 \
list-notebook-executions --from 1593400000.000

{
    "NotebookExecutions": [
        {
            "NotebookExecutionId": "ex-IZWZX78UVPAATC8LHJR129B1RBN4T",
            "EditorId": "e-BKTM2DIHXBEDRU44ANWRKIU8N",
            "NotebookExecutionName": "my-execution",
            "Status": "STOPPED",
            "StartTime": 1593490876.241
        },
        {
            "NotebookExecutionId": "ex-IZWZZVR9DKQ9WQ7VZWXJZR29UGHTE",
            "EditorId": "e-BKTM2DIHXBEDRU44ANWRKIU8N",
            "NotebookExecutionName": "my-execution",
            "Status": "RUNNING",
            "StartTime": 1593490857.009
        },
        {
            "NotebookExecutionId": "ex-IZWZYRS0M14L5V95WZ9OQ399SKMNW",
            "EditorId": "e-BKTM2DIHXBEDRU44ANWRKIU8N",
            "NotebookExecutionName": "my-execution",
            "Status": "STOPPED",
            "StartTime": 1593490292.995
        },
        {
            "NotebookExecutionId": "ex-IZX009ZK83IVY5E33VH8MDMELVK8K",
            "EditorId": "e-BKTM2DIHXBEDRU44ANWRKIU8N",
            "NotebookExecutionName": "my-execution",
            "Status": "FINISHED",
```

```
            "StartTime": 1593489834.765
        },
        {
            "NotebookExecutionId": "ex-IZWZXOZF88JWDF9J09GJ91R57VI0N",
            "EditorId": "e-BKTM2DIHXBEDRU44ANWRKIU8N",
            "NotebookExecutionName": "my-execution",
            "Status": "FAILED",
            "StartTime": 1593488934.688
        }
    ]
}
```

Buat daftar eksekusi untuk buku catatan berdasarkan waktu mulai dan status

`list-notebook-executions`Perintah juga dapat mengambil `--status` parameter untuk memfilter hasil.

```
# filter by start time and status
aws emr --region us-east-1 \
list-notebook-executions --from 1593400000.000 --status FINISHED
{
    "NotebookExecutions": [
        {
            "NotebookExecutionId": "ex-IZWZZVR9DKQ9WQ7VZWXJZR29UGHTE",
            "EditorId": "e-BKTM2DIHXBEDRU44ANWRKIU8N",
            "NotebookExecutionName": "my-execution",
            "Status": "FINISHED",
            "StartTime": 1593490857.009
        },
        {
            "NotebookExecutionId": "ex-IZX009ZK83IVY5E33VH8MDMELVK8K",
            "EditorId": "e-BKTM2DIHXBEDRU44ANWRKIU8N",
            "NotebookExecutionName": "my-execution",
            "Status": "FINISHED",
            "StartTime": 1593489834.765
        }
    ]
}
```

Sampel eksekusi notebook Python

> **Note**
>
> EMR Notebooks tersedia sebagai EMR Studio Workspaces di konsol baru. Anda masih dapat menggunakan notebook yang ada di konsol lama, tetapi Anda tidak dapat membuat notebook baru di konsol lama. Tombol Create Workspace di konsol baru menggantikan fungsi ini. Untuk mengakses atau membuat Ruang Kerja, pengguna EMR Notebooks memerlukan izin peran IAM tambahan. Untuk informasi selengkapnya, lihat Amazon EMR Notebook adalah Amazon EMR Studio Workspaces di konsol baru dan Apa yang baru di konsol?

Contoh kode berikut adalah file SDK for Python (Boto3) bernama `demo.py` yang menunjukkan API eksekusi notebook.

Untuk informasi tentang tindakan Amazon EMR API, lihat `NotebookExecution` tindakan Amazon EMR API.

```
import boto3,time

emr = boto3.client(
    'emr',
    region_name='us-west-1'
)

start_resp = emr.start_notebook_execution(
    EditorId='e-40AC8ZO6EGGCPJ4DLO48KGGGI',
    RelativePath='boto3_demo.ipynb',
    ExecutionEngine={'Id':'j-1HYZS6JQKV11Q'},
    ServiceRole='EMR_Notebooks_DefaultRole'
)

execution_id = start_resp["NotebookExecutionId"]
print(execution_id)
print("\n")

describe_response = emr.describe_notebook_execution(NotebookExecutionId=execution_id)

print(describe_response)
print("\n")
```

```
list_response = emr.list_notebook_executions()
print("Existing notebook executions:\n")
for execution in list_response['NotebookExecutions']:
    print(execution)
    print("\n")

print("Sleeping for 5 sec...")
time.sleep(5)

print("Stop execution " + execution_id)
emr.stop_notebook_execution(NotebookExecutionId=execution_id)
describe_response = emr.describe_notebook_execution(NotebookExecutionId=execution_id)
print(describe_response)
print("\n")
```

Berikut output dari menjalankan`demo.py`.

```
ex-IZX56YJDW1D29Q1PHR32WABU2SAPK

{'NotebookExecution': {'NotebookExecutionId': 'ex-IZX56YJDW1D29Q1PHR32WABU2SAPK',
 'EditorId': 'e-40AC8ZO6EGGCPJ4DLO48KGGGI', 'ExecutionEngine': {'Id':
 'j-1HYZS6JQKV11Q', 'Type': 'EMR'}, 'NotebookExecutionName': '', 'Status': 'STARTING',
 'StartTime': datetime.datetime(2020, 8, 19, 0, 49, 19, 418000, tzinfo=tzlocal()),
 'Arn': 'arn:aws:elasticmapreduce:us-west-1:123456789012:notebook-execution/ex-
IZX56YJDW1D29Q1PHR32WABU2SAPK', 'LastStateChangeReason': 'Execution is starting
 for cluster j-1HYZS6JQKV11Q.', 'Tags': []}, 'ResponseMetadata': {'RequestId':
 '70f12c5f-1dda-45b7-adf6-964987d373b7', 'HTTPStatusCode': 200, 'HTTPHeaders': {'x-
amzn-requestid': '70f12c5f-1dda-45b7-adf6-964987d373b7', 'content-type': 'application/
x-amz-json-1.1', 'content-length': '448', 'date': 'Wed, 19 Aug 2020 00:49:22 GMT'},
 'RetryAttempts': 0}}

Existing notebook executions:

{'NotebookExecutionId': 'ex-IZX56YJDW1D29Q1PHR32WABU2SAPK', 'EditorId':
 'e-40AC8ZO6EGGCPJ4DLO48KGGGI', 'NotebookExecutionName': '', 'Status': 'STARTING',
 'StartTime': datetime.datetime(2020, 8, 19, 0, 49, 19, 418000, tzinfo=tzlocal())}


{'NotebookExecutionId': 'ex-IZX5ABS5PR1E5AHMFYEMX3JJIORRB', 'EditorId':
 'e-40AC8ZO6EGGCPJ4DLO48KGGGI', 'NotebookExecutionName': '', 'Status': 'RUNNING',
 'StartTime': datetime.datetime(2020, 8, 19, 0, 48, 36, 373000, tzinfo=tzlocal())}
```

```
{'NotebookExecutionId': 'ex-IZX5GLVXIU1HNI8BWVW057F6MF4VE', 'EditorId':
 'e-40AC8ZO6EGGCPJ4DLO48KGGGI', 'NotebookExecutionName': '', 'Status': 'FINISHED',
 'StartTime': datetime.datetime(2020, 8, 19, 0, 45, 14, 646000, tzinfo=tzlocal()),
 'EndTime': datetime.datetime(2020, 8, 19, 0, 46, 26, 543000, tzinfo=tzlocal())}


{'NotebookExecutionId': 'ex-IZX5CV8YDUO8JAIWMXN2VH32RUIT1', 'EditorId':
 'e-40AC8ZO6EGGCPJ4DLO48KGGGI', 'NotebookExecutionName': '', 'Status': 'FINISHED',
 'StartTime': datetime.datetime(2020, 8, 19, 0, 43, 5, 807000, tzinfo=tzlocal()),
 'EndTime': datetime.datetime(2020, 8, 19, 0, 44, 31, 632000, tzinfo=tzlocal())}


{'NotebookExecutionId': 'ex-IZX5AS0PPW55CEDEURZ9NSOWSUJZ6', 'EditorId':
 'e-40AC8ZO6EGGCPJ4DLO48KGGGI', 'NotebookExecutionName': '', 'Status': 'FINISHED',
 'StartTime': datetime.datetime(2020, 8, 19, 0, 42, 29, 265000, tzinfo=tzlocal()),
 'EndTime': datetime.datetime(2020, 8, 19, 0, 43, 48, 320000, tzinfo=tzlocal())}


{'NotebookExecutionId': 'ex-IZX57YF5Q53BKWLR4I5QZ14HJ7DRS', 'EditorId':
 'e-40AC8ZO6EGGCPJ4DLO48KGGGI', 'NotebookExecutionName': '', 'Status': 'FINISHED',
 'StartTime': datetime.datetime(2020, 8, 19, 0, 38, 37, 81000, tzinfo=tzlocal()),
 'EndTime': datetime.datetime(2020, 8, 19, 0, 40, 39, 646000, tzinfo=tzlocal())}

Sleeping for 5 sec...
Stop execution ex-IZX56YJDW1D29Q1PHR32WABU2SAPK
{'NotebookExecution': {'NotebookExecutionId': 'ex-IZX56YJDW1D29Q1PHR32WABU2SAPK',
 'EditorId': 'e-40AC8ZO6EGGCPJ4DLO48KGGGI', 'ExecutionEngine': {'Id':
 'j-1HYZS6JQKV11Q', 'Type': 'EMR'}, 'NotebookExecutionName': '', 'Status': 'STOPPING',
 'StartTime': datetime.datetime(2020, 8, 19, 0, 49, 19, 418000, tzinfo=tzlocal()),
 'Arn': 'arn:aws:elasticmapreduce:us-west-1:123456789012:notebook-execution/ex-
IZX56YJDW1D29Q1PHR32WABU2SAPK', 'LastStateChangeReason': 'Execution is being stopped
 for cluster j-1HYZS6JQKV11Q.', 'Tags': []}, 'ResponseMetadata': {'RequestId':
 '2a77ef73-c1c6-467c-a1d1-7204ab2f6a53', 'HTTPStatusCode': 200, 'HTTPHeaders': {'x-
amzn-requestid': '2a77ef73-c1c6-467c-a1d1-7204ab2f6a53', 'content-type': 'application/
x-amz-json-1.1', 'content-length': '453', 'date': 'Wed, 19 Aug 2020 00:49:30 GMT'},
 'RetryAttempts': 0}}
```

Sampel eksekusi notebook Ruby

> **Note**
>
> EMR Notebooks tersedia sebagai EMR Studio Workspaces di konsol baru. Anda masih dapat menggunakan notebook yang ada di konsol lama, tetapi Anda tidak dapat membuat notebook baru di konsol lama. Tombol Create Workspace di konsol baru menggantikan fungsi ini. Untuk mengakses atau membuat Ruang Kerja, pengguna EMR Notebooks memerlukan izin peran IAM tambahan. Untuk informasi selengkapnya, lihat Amazon EMR Notebook adalah Amazon EMR Studio Workspaces di konsol baru dan Apa yang baru di konsol?

Berikut ini adalah sampel kode Ruby yang menunjukkan menggunakan API eksekusi notebook.

```
# prepare an Amazon EMR client

emr = Aws::EMR::Client.new(
  region: 'us-east-1',
  access_key_id: 'AKIA...JKPKA',
  secret_access_key: 'rLMeu...vU0OLrAC1',
)
```

Memulai eksekusi notebook dan mendapatkan id eksekusi

Dalam contoh ini, editor Amazon S3 dan notebook EMR adalah. `s3://mybucket/notebooks/e-EA8VGAA429FEQTC8HC9ZHWISK/test.ipynb`

Untuk informasi tentang tindakan Amazon EMR API, lihat `NotebookExecution` tindakan Amazon EMR API.

```
start_response = emr.start_notebook_execution({
    editor_id: "e-EA8VGAA429FEQTC8HC9ZHWISK",
    relative_path: "test.ipynb",

    execution_engine: {id: "j-3U82I95AMALGE"},

    service_role: "EMR_Notebooks_DefaultRole",
})
```

```
notebook_execution_id = start_resp.notebook_execution_id
```

Menggambarkan eksekusi notebook dan mencetak detailnya

```
describe_resp = emr.describe_notebook_execution({
    notebook_execution_id: notebook_execution_id
})
puts describe_resp.notebook_execution
```

Output dari perintah di atas adalah sebagai berikut.

```
{
:notebook_execution_id=>"ex-IZX3VTVZWVWPP27KUB90BZ7V9IEDG",
:editor_id=>"e-EA8VGAA429FEQTC8HC9ZHWISK",
:execution_engine=>{:id=>"j-3U82I95AMALGE", :type=>"EMR", :master_instance_security_group_id=>n
:notebook_execution_name=>"",
:notebook_params=>nil,
:status=>"STARTING",
:start_time=>2020-07-23 15:07:07 -0700,
:end_time=>nil,
:arn=>"arn:aws:elasticmapreduce:us-east-1:123456789012:notebook-execution/ex-
IZX3VTVZWVWPP27KUB90BZ7V9IEDG",
:output_notebook_uri=>nil,
:last_state_change_reason=>"Execution is starting for cluster
 j-3U82I95AMALGE.", :notebook_instance_security_group_id=>nil,
:tags=>[]
}
```

Filter notebook

```

"EditorId": "e-XXXX",           [Optional]
"From" : "1593400000.000",    [Optional]
"To" :
```

Menghentikan eksekusi notebook

```
stop_resp = emr.stop_notebook_execution({
    notebook_execution_id: notebook_execution_id
})
```

Mengaktifkan peniruan pengguna untuk memantau aktivitas pengguna dan tugas Spark

> **Note**
>
> EMR Notebooks tersedia sebagai EMR Studio Workspaces di konsol baru. Anda masih dapat menggunakan notebook yang ada di konsol lama, tetapi Anda tidak dapat membuat notebook baru di konsol lama. Tombol Create Workspace di konsol baru menggantikan fungsi ini. Untuk mengakses atau membuat Ruang Kerja, pengguna EMR Notebooks memerlukan izin peran IAM tambahan. Untuk informasi selengkapnya, lihat Amazon EMR Notebook adalah Amazon EMR Studio Workspaces di konsol baru dan Apa yang baru di konsol?

EMR Notebooks memungkinkan Anda untuk mengonfigurasi peniruan pengguna pada klaster Spark. Fitur ini membantu Anda melacak aktivitas tugas yang dimulai dari dalam editor notebook. Selain itu, EMR Notebooks memiliki widget Jupyter Notebook bawaan untuk melihat detail tugas Spark bersama output kueri di editor notebook. Widget ini tersedia secara default dan tidak memerlukan konfigurasi khusus. Namun, untuk melihat server riwayat, klien Anda harus dikonfigurasi untuk melihat antarmuka web Amazon EMR yang di-host di node utama.

Menyiapkan peniruan pengguna Spark

Secara default, tugas Spark yang dikirimkan pengguna menggunakan editor notebook tampaknya berasal dari identitas pengguna `livy`. Anda dapat mengonfigurasi peniruan identitas pengguna untuk klaster sehingga pekerjaan ini terkait dengan identitas pengguna yang menjalankan kode sebagai gantinya. Direktori pengguna HDFS pada node utama dibuat untuk setiap identitas pengguna yang menjalankan kode di notebook. Misalnya, jika pengguna `NbUser1` menjalankan kode dari editor notebook, Anda dapat terhubung ke node utama dan melihat yang `hadoop fs -ls / user` menunjukkan direktori`/user/user_NbUser1`.

Anda mengaktifkan fitur ini dengan menetapkan properti di klasifikasi konfigurasi `core-site` dan `livy-conf`. Fitur ini tidak tersedia secara default ketika Anda meminta Amazon EMR membuat klaster bersama dengan notebook. Untuk informasi selengkapnya tentang menggunakan klasifikasi untuk mengustomisasi aplikasi, lihat Mengonfigurasi aplikasi dalam Panduan Rilis Amazon EMR.

Gunakan klasifikasi konfigurasi berikut dan nilai-nilai untuk mengaktifkan peniruan pengguna untuk EMR Notebooks:

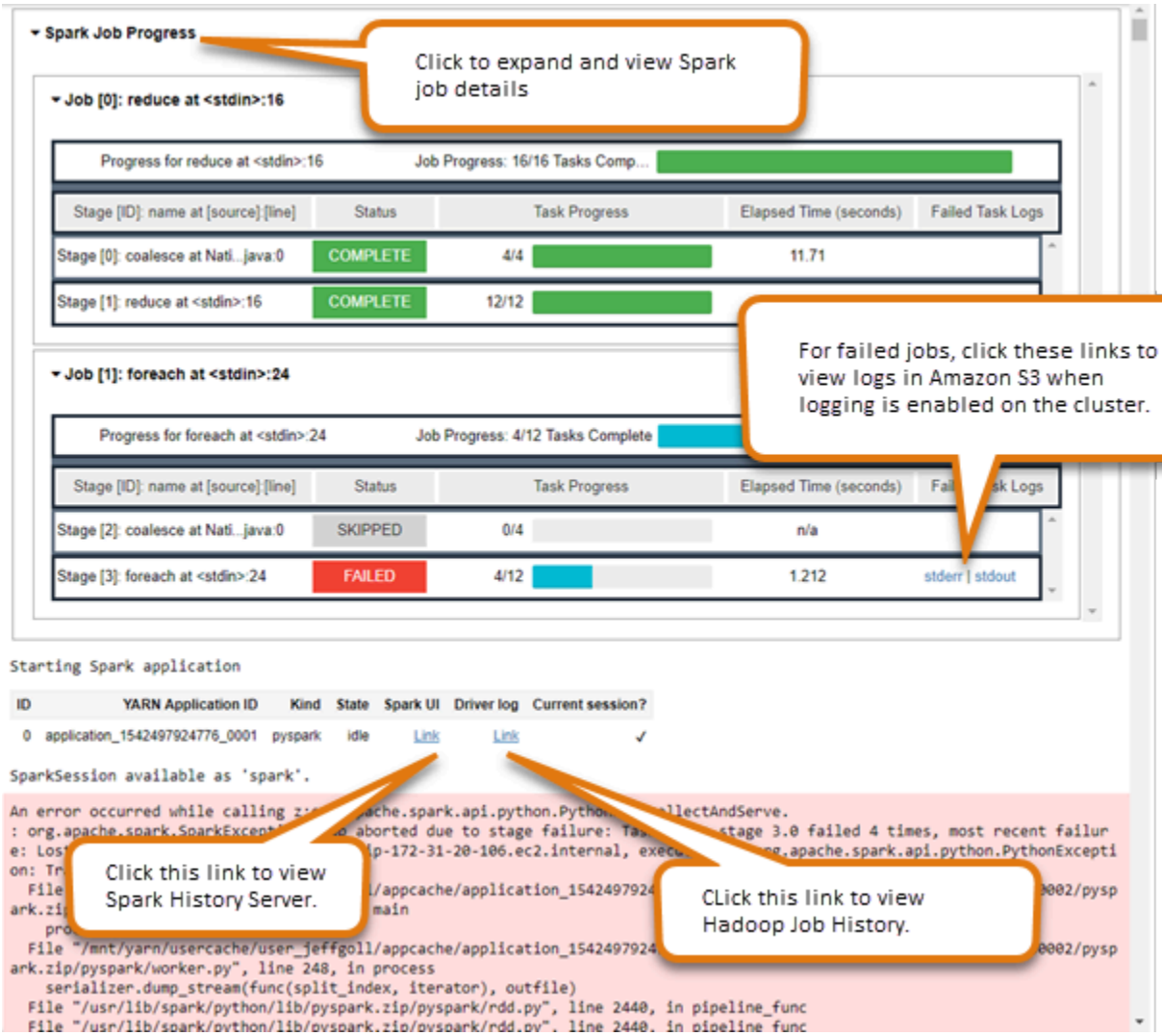
```
[
    {
        "Classification": "core-site",
        "Properties": {
          "hadoop.proxyuser.livy.groups": "*",
          "hadoop.proxyuser.livy.hosts": "*"
        }
    },
    {
        "Classification": "livy-conf",
        "Properties": {
          "livy.impersonation.enabled": "true"
        }
    }
]
```

Menggunakan widget pemantauan tugas Spark

Ketika Anda menjalankan kode dalam editor notebook yang mengeksekusi tugas Spark pada klaster EMR, output termasuk widget Jupyter Notebook untuk pemantauan tugas Spark. Widget memberikan detail tugas dan tautan yang berguna ke halaman server riwayat Spark dan halaman riwayat tugas Hadoop, bersama dengan tautan yang nyaman untuk log tugas di Amazon S3 untuk tugas gagal.

Untuk melihat halaman server riwayat pada node utama cluster, Anda harus mengatur klien SSH dan proxy yang sesuai. Untuk informasi selengkapnya, lihat Melihat antarmuka web yang di-host pada klaster Amazon EMR. Untuk melihat log di Amazon S3, pencatatan klaster harus diaktifkan, yang merupakan default untuk klaster baru. Untuk informasi selengkapnya, lihat Melihat berkas log yang diarsipkan ke Amazon S3.

Berikut ini adalah contoh dari pemantauan tugas Spark.



Keamanan dan kontrol akses EMR notebooks

> **Note**
>
> EMR Notebooks tersedia sebagai EMR Studio Workspaces di konsol baru. Anda masih dapat menggunakan notebook yang ada di konsol lama, tetapi Anda tidak dapat membuat notebook baru di konsol lama. Tombol Create Workspace di konsol baru menggantikan fungsi ini. Untuk mengakses atau membuat Ruang Kerja, pengguna EMR Notebooks memerlukan izin peran IAM tambahan. Untuk informasi selengkapnya, lihat Amazon EMR Notebook adalah Amazon EMR Studio Workspaces di konsol baru dan Apa yang baru di konsol?

Beberapa fitur tersedia untuk membantu Anda menyesuaikan postur keamanan EMR Notebooks. Hal ini membantu memastikan bahwa hanya pengguna yang berwenang memiliki akses ke EMR notebook, dapat bekerja dengan notebook, dan menggunakan editor notebook untuk mengeksekusi kode pada klaster. Fitur-fitur ini bekerja bersama dengan fitur keamanan yang tersedia untuk klaster Amazon EMR dan Amazon EMR. Untuk informasi selengkapnya, lihat Keamanan di Amazon EMR.

- Anda dapat menggunakan pernyataan kebijakan AWS Identity and Access Management kebijakan bersama dengan tanda notebook untuk membatasi akses. Untuk informasi lebih lanjut, lihat Cara kerja Amazon EMR dengan IAM dan Contoh pernyataan kebijakan berbasis identitas untuk EMR Notebooks.
- Grup keamanan Amazon EC2 bertindak sebagai firewall virtual yang mengontrol lalu lintas jaringan antara instance utama cluster dan editor notebook. Anda dapat menggunakan default atau kustomisasi grup keamanan ini. Untuk informasi selengkapnya, lihat Menentukan grup-grup keamanan EC2 untuk EMR Notebooks.
- Anda menentukan Peran Layanan AWS yang menentukan izin apa yang dimiliki EMR notebook saat berinteraksi dengan layanan AWS lain. Untuk informasi selengkapnya, lihat Peran layanan untuk EMR Notebooks.

Memasang dan menggunakan kernel dan pustaka

> **Note**
>
> EMR Notebooks tersedia sebagai EMR Studio Workspaces di konsol baru. Anda masih dapat menggunakan notebook yang ada di konsol lama, tetapi Anda tidak dapat membuat notebook baru di konsol lama. Tombol Create Workspace di konsol baru menggantikan fungsi ini. Untuk mengakses atau membuat Ruang Kerja, pengguna EMR Notebooks memerlukan izin peran IAM tambahan. Untuk informasi selengkapnya, lihat Amazon EMR Notebook adalah Amazon EMR Studio Workspaces di konsol baru dan Apa yang baru di konsol?

Setiap EMR notebook dilengkapi dengan satu set perpustakaan dan kernel pra-instal. Anda dapat menginstal pustaka dan kernel tambahan di kluster EMR jika cluster memiliki akses ke repositori tempat kernel dan pustaka berada. Misalnya, untuk klaster di subnet privat, Anda mungkin perlu mengonfirmasi terjemahan alamat jaringan (NAT) dan menyediakan jalur bagi klaster untuk mengakses repositori PyPI publik untuk menginstal perpustakaan. Untuk informasi lebih lanjut

tentang konfigurasi akses eksternal untuk konfigurasi jaringan yang berbeda, lihat Skenario dan contoh di Panduan Pengguna Amazon VPC.

Aplikasi EMR Tanpa Server dilengkapi dengan pustaka pra-instal berikut untuk Python dan: PySpark

- Pustaka Python —ggplot,,matplotlib,,numpy,pandas,plotly, bokeh scikit-learn scipy scipy
- PySpark perpustakaan —ggplot,,matplotlib,numpy,pandas,plotly,bokeh,scikit-learn, scipy scipy

Menginstal kernel dan pustaka Python pada node primer cluster

Dengan versi rilis Amazon EMR 5.30.0 dan yang lebih baru, tidak termasuk 6.0.0, Anda dapat menginstal pustaka dan kernel Python tambahan pada node utama cluster. Setelah instalasi, kernel dan perpustakaan ini tersedia untuk setiap pengguna yang menjalankan EMR notebook yang melekat pada klaster. Pustaka Python yang diinstal dengan cara ini hanya tersedia untuk proses yang berjalan pada node utama. Perpustakaan tidak diinstal pada simpul inti atau tugas dan tidak tersedia untuk eksekutor yang berjalan pada simpul tersebut.

> **Note**
>
> Untuk Amazon EMR versi 5.30.1, 5.31.0, dan 6.1.0, Anda harus mengambil langkah-langkah tambahan untuk menginstal kernel dan pustaka pada node utama cluster.
> Untuk mengaktifkan fitur, lakukan hal berikut ini:
>
> 1. Pastikan bahwa kebijakan izin yang dilampirkan ke peran layanan untuk EMR Notebooks mengizinkan tindakan berikut ini:
>
>    ```
>    elasticmapreduce:ListSteps
>    ```
>
>   Untuk informasi selengkapnya, lihat Peran layanan untuk EMR Notebooks.
> 2. Gunakan AWS CLI untuk menjalankan langkah pada klaster yang mengatur EMR Notebooks seperti yang ditunjukkan dalam contoh berikut. Anda harus menggunakan nama langkah `EMRNotebooksSetup`. Ganti *`us-east-1`* dengan Wilayah di mana klaster Anda berada. Untuk informasi selengkapnya, lihat Menambahkan langkah-langkah untuk klaster menggunakan AWS CLI.
>
>    ```
>    aws emr add-steps --cluster-id MyClusterID --steps
>     Type=CUSTOM_JAR,Name=EMRNotebooksSetup,ActionOnFailure=CONTINUE,Jar=s3://us-
>    east-1.elasticmapreduce/libs/script-runner/script-runner.jar,Args=["s3://
>    ```

```
awssupportdatasvcs.com/bootstrap-actions/EMRNotebooksSetup/emr-notebooks-
setup.sh"]
```

Anda dapat menginstal kernel dan pustaka menggunakan `pip` atau `conda` di `/emr/notebook-env/bin` direktori pada node utama.

Example — Menginstal pustaka Python

Dari kernel Python3, jalankan `%pip` sihir sebagai perintah dari dalam sel notebook untuk menginstal pustaka Python.

```
%pip install pmdarima
```

Anda mungkin perlu me-restart kernel untuk menggunakan paket yang diperbarui. Anda juga dapat menggunakan sihir `%%sh`Spark untuk memanggil`pip`.

```
%%sh
/emr/notebook-env/bin/pip install -U matplotlib
/emr/notebook-env/bin/pip install -U pmdarima
```

Saat menggunakan PySpark kernel, Anda dapat menginstal pustaka di cluster menggunakan `pip` perintah atau menggunakan pustaka dengan cakupan notebook dari dalam buku catatan. PySpark

Untuk menjalankan `pip` perintah pada cluster dari terminal, pertama-tama hubungkan ke node utama menggunakan SSH, seperti yang ditunjukkan oleh perintah berikut.

```
sudo pip3 install -U matplotlib
sudo pip3 install -U pmdarima
```

Atau, Anda dapat menggunakan pustaka dengan cakupan notebook. Dengan pustaka dengan cakupan notebook, instalasi perpustakaan Anda terbatas pada cakupan sesi Anda dan terjadi pada semua pelaksana Spark. Untuk informasi selengkapnya, lihat Menggunakan Pustaka Cakupan Notebook.

Jika Anda ingin mengemas beberapa pustaka Python dalam PySpark kernel, Anda juga dapat membuat lingkungan virtual Python yang terisolasi. Sebagai contoh, lihat Menggunakan Virtualenv.

Untuk membuat lingkungan virtual Python dalam sesi, gunakan properti Spark `spark.yarn.dist.archives` dari perintah `%%configure` ajaib di sel pertama dalam buku catatan, seperti contoh berikut menunjukkan.

```
%%configure -f
{
   "conf": {
   "spark.yarn.appMasterEnv.PYSPARK_PYTHON":"./environment/bin/python",
   "spark.yarn.appMasterEnv.PYSPARK_DRIVER_PYTHON":"./environment/bin/python",
   "spark.yarn.dist.archives":"s3://DOC-EXAMPLE-BUCKET/prefix/
my_pyspark_venv.tar.gz#environment",
   "spark.submit.deployMode":"cluster"
   }
}
```

Anda juga dapat membuat lingkungan pelaksana Spark.

```
%%configure -f
{
   "conf": {
   "spark.yarn.appMasterEnv.PYSPARK_PYTHON":"./environment/bin/python",
   "spark.yarn.appMasterEnv.PYSPARK_DRIVER_PYTHON":"./environment/bin/python",
   "spark.executorEnv.PYSPARK_PYTHON":"./environment/bin/python",
   "spark.yarn.dist.archives":"s3://DOC-EXAMPLE-BUCKET/prefix/
my_pyspark_venv.tar.gz#environment",
   "spark.submit.deployMode":"cluster"
   }
}
```

Anda juga dapat menggunakan `conda` untuk menginstal pustaka Python. Anda tidak perlu akses sudo untuk menggunakannya`conda`. Anda harus terhubung ke node utama dengan SSH, dan kemudian jalankan `conda` dari terminal. Untuk informasi selengkapnya, lihat Connect ke node utama menggunakan SSH.

Example — Memasang kernel

Contoh berikut menunjukkan penginstalan kernel Kotlin menggunakan perintah terminal saat terhubung ke node utama klaster:

```
sudo /emr/notebook-env/bin/conda install kotlin-jupyter-kernel -c jetbrains
```

> **Note**
>
> Instruksi ini tidak menginstal dependensi kernel. Jika kernel Anda memiliki dependensi pihak ketiga, Anda mungkin perlu mengambil langkah penyiapan tambahan sebelum dapat menggunakan kernel dengan notebook Anda.

Pertimbangan dan batasan dengan pustaka cakupan notebook

Saat Anda menggunakan pustaka dengan cakupan notebook, pertimbangkan hal berikut:

- Pustaka dengan cakupan notebook tersedia untuk kluster yang Anda buat dengan rilis Amazon EMR 5.26.0 dan yang lebih tinggi.
- Pustaka dengan cakupan notebook dimaksudkan untuk digunakan hanya dengan kernel. PySpark
- Setiap pengguna dapat menginstal pustaka cakupan notebook tambahan dari dalam sel notebook. Pustaka ini hanya tersedia untuk pengguna notebook tersebut selama sesi notebook tunggal. Jika pengguna lain membutuhkan pustaka yang sama, atau pengguna yang sama membutuhkan pustaka yang sama dalam sesi yang berbeda, pustaka harus diinstal ulang.
- Anda hanya dapat menghapus pustaka yang diinstal dengan API. `install_pypi_package` Anda tidak dapat menghapus pustaka apa pun yang telah diinstal sebelumnya di cluster.
- Jika pustaka yang sama dengan versi yang berbeda diinstal pada klaster dan sebagai pustaka cakupan notebook, versi pustaka cakupan notebook menimpa versi pustaka klaster.

Bekerja dengan Pustaka cakupan notebook

Untuk menginstal pustaka, klaster Amazon EMR Anda harus memiliki akses ke repositori PyPI di mana pustaka berada.

Contoh berikut menunjukkan perintah sederhana untuk membuat daftar, menginstal, dan menghapus pustaka dari dalam sel notebook menggunakan PySpark kernel dan API. Untuk contoh tambahan, lihat posting Instal pustaka Python pada sebuah klaster yang sedang berjalan dengan EMR Notebooks di AWS Blog Big Data.

Example — Daftar pustaka saat ini

Perintah berikut membuat daftar paket Python yang tersedia untuk sesi notebook Spark saat ini. Ini berisi daftar pustaka yang diinstal pada klaster dan pustaka cakupan notebook.

```
sc.list_packages()
```

Example — Menginstal pustaka Celery

Perintah berikut menginstal pustaka Celery sebagai pustaka cakupan notebook.

```
sc.install_pypi_package("celery")
```

Setelah menginstal pustaka, perintah berikut mengonfirmasi bahwa pustaka tersedia pada driver dan eksekutor Spark.

```
import celery
sc.range(1,10000,1,100).map(lambda x: celery.__version__).collect()
```

Example — Menginstal pustaka Arrow, menentukan versi dan repositori

Perintah berikut menginstal pustaka Arrow sebagai pustaka notebook, dengan spesifikasi versi pustaka dan URL repositori.

```
sc.install_pypi_package("arrow==0.14.0", "https://pypi.org/simple")
```

Example — Menghapus instalasi pustaka

Perintah berikut menghapus instalan pustaka Arrow, menghapusnya sebagai pustaka cakupan notebook dari sesi saat ini.

```
sc.uninstall_package("arrow")
```

Mengasosiasikan repositori berbasis Git dengan EMR Notebooks

> **Note**
>
> EMR Notebooks tersedia sebagai EMR Studio Workspaces di konsol baru. Anda masih dapat menggunakan notebook yang ada di konsol lama, tetapi Anda tidak dapat membuat notebook baru di konsol lama. Tombol Create Workspace di konsol baru menggantikan fungsi ini. Untuk mengakses atau membuat Ruang Kerja, pengguna EMR Notebooks memerlukan izin peran IAM tambahan. Untuk informasi selengkapnya, lihat Amazon EMR Notebook adalah Amazon EMR Studio Workspaces di konsol baru dan Apa yang baru di konsol?

Anda dapat mengasosiasikan repositori berbasis Git dengan Amazon EMR notebooks untuk menyimpan notebook Anda dalam lingkungan terkendali versi. Anda dapat mengasosiasikan hingga tiga repositori dengan notebook. Layanan berbasis Git berikut didukung:

- AWS CodeCommit
- GitHub
- Bitbucket
- GitLab

Mengasosiasikan repositori berbasis Git dengan notebook Anda memiliki manfaat sebagai berikut.

- Kontrol versi — Anda dapat merekam perubahan kode dalam sistem kontrol versi sehingga Anda dapat meninjau riwayat perubahan Anda dan secara selektif membalikkannya.
- Kolaborasi — Rekan kerja yang bekerja di notebook berbeda dapat berbagi kode melalui repositori berbasis Git jarak jauh. Notebook dapat mengkloning atau menggabungkan kode dari repositori jarak jauh dan mendorong perubahan kembali ke repositori jarak jauh tersebut.
- Penggunaan kembali kode — Banyak notebook Jupyter yang menunjukkan analisis data atau teknik pembelajaran mesin tersedia di repositori yang dihosting publik, seperti. GitHub Anda dapat mengasosiasikan notebook Anda dengan repositori untuk menggunakan kembali notebook Jupyter yang berada dalam repositori.

Untuk menggunakan repositori berbasis Git dengan EMR Notebooks, Anda menambahkan repositori sebagai sumber daya di konsol Amazon EMR, mengasosiasikan kredensial untuk repositori yang memerlukan autentikasi, dan mengaitkannya dengan notebook Anda. Anda dapat melihat daftar repositori yang disimpan di akun Anda dan detail tentang setiap repositori di konsol Amazon EMR. Anda dapat mengasosiasikan repositori berbasis Git yang ada dengan notebook saat Anda membuatnya.

Topik

- Prasyarat dan pertimbangan
- Tambahkan repositori berbasis Git ke Amazon EMR
- Memperbarui atau menghapus repositori berbasis Git
- Tautkan atau hapus tautan repositori berbasis Git
- Buat Notebook baru dengan repositori Git terkait
- Gunakan repositori Git di Notebook

Prasyarat dan pertimbangan

> **Note**
>
> EMR Notebooks tersedia sebagai EMR Studio Workspaces di konsol baru. Anda masih dapat menggunakan notebook yang ada di konsol lama, tetapi Anda tidak dapat membuat notebook baru di konsol lama. Tombol Create Workspace di konsol baru menggantikan fungsi ini. Untuk mengakses atau membuat Ruang Kerja, pengguna EMR Notebooks memerlukan izin peran IAM tambahan. Untuk informasi selengkapnya, lihat Amazon EMR Notebook adalah Amazon EMR Studio Workspaces di konsol baru dan Apa yang baru di konsol?

Pertimbangkan hal berikut saat merencanakan untuk mengintegrasikan repositori berbasis Git dengan EMR Notebooks.

AWS CodeCommit

Jika Anda menggunakan CodeCommit repositori, Anda harus menggunakan kredensi Git dan HTTPS dengan. CodeCommit SSH Keys, dan HTTPS dengan pembantu kredensial AWS CLI tidak didukung. CodeCommit tidak mendukung token akses pribadi (PATs). Untuk informasi selengkapnya, lihat Menggunakan IAM dengan CodeCommit: Kredensyal Git, kunci SSH, dan kunci AWS akses di Panduan Pengguna IAM dan Pengaturan untuk pengguna HTTPS yang menggunakan kredensyal Git di Panduan Pengguna. AWS CodeCommit

Pertimbangan akses dan izin

Sebelum mengasosiasikan repositori dengan notebook Anda, pastikan bahwa klaster Anda, IAM role untuk EMR Notebooks, dan grup keamanan memiliki pengaturan dan izin yang benar. Anda juga dapat mengonfigurasi repositori berbasis Git yang Anda host di jaringan privat dengan mengikuti petunjuk di Mengonfigurasi repositori Git yang di-host secara privat untuk EMR Notebooks.

- Akses internet klaster — Antarmuka jaringan yang diluncurkan hanya memiliki alamat IP pribadi. Ini berarti bahwa klaster yang menghubungkan notebook Anda harus dalam subnet privat dengan gateway terjemahan alamat jaringan (NAT) atau harus dapat mengakses internet melalui virtual private gateway. Untuk informasi selengkapnya, lihat Opsi Amazon VPC?.

  Grup keamanan untuk notebook Anda harus menyertakan aturan keluar yang memungkinkan notebook untuk mengarahkan lalu lintas ke internet dari klaster. Kami menyarankan agar Anda

membuat grup keamanan Anda sendiri. Untuk informasi lebih lanjut, lihat Menentukan grup keamanan EC2 untuk EMR Notebooks.

> **Important**
>
> Jika antarmuka jaringan diluncurkan ke subnet publik, antarmuka tersebut tidak akan dapat berkomunikasi dengan internet melalui gateway internet (IGW).

- Izin untuk AWS Secrets Manager — Jika Anda menggunakan Secrets Manager untuk menyimpan rahasia yang Anda gunakan untuk mengakses repositori, the section called "Peran EMR Notebooks" harus memiliki kebijakan izin terlampir yang memungkinkan tindakan `secretsmanager:GetSecretValue`.

Mengonfigurasi repositori Git yang di-host secara privat untuk EMR Notebooks

Gunakan petunjuk berikut untuk mengonfigurasi repositori yang dihost secara privat untuk EMR Notebooks. Anda harus menyediakan file konfigurasi dengan informasi tentang server DNS dan Git Anda. Amazon EMR menggunakan informasi ini untuk mengonfigurasi EMR notebook yang dapat merutekan lalu lintas ke repositori yang Anda host secara privat.

Prasyarat

Sebelum Anda mengonfigurasi repositori Git yang di-host secara privat untuk EMR Notebooks, Anda harus memiliki yang berikut:

- Amazon S3 ControlLokasi tempat file untuk notebook EMR Anda akan disimpan.

Untuk mengonfigurasi satu atau beberapa repositori Git yang di-host secara privat untuk EMR Notebooks

1. Buat file konfigurasi menggunakan templat yang disediakan. Sertakan nilai berikut untuk setiap server Git yang ingin Anda tentukan dalam konfigurasi Anda:

   - **`DnsServerIpV4`** - Alamat IPv4 dari server DNS Anda. Jika Anda memberikan nilai untuk `DnsServerIpV4` dan `GitServerIpV4List`, nilai untuk `DnsServerIpV4` diutamakan dan akan digunakan untuk menyelesaikan `GitServerDnsName` Anda.

> **Note**
>
> Untuk menggunakan repositori Git yang di-host secara privat, server DNS Anda harus mengizinkan akses masuk dari EMR Notebooks. Kami sangat menyarankan Anda mengamankan server DNS Anda terhadap akses tidak sah lainnya.

- **`GitServerDnsName`** - Nama DNS server Git Anda. Sebagai contoh `"git.example.com"`.
- **`GitServerIpV4List`** - Daftar alamat IPv4 milik server Git Anda.

```
[
    {
        "Type": "PrivatelyHostedGitConfig",
        "Value": [
            {
                "DnsServerIpV4": "<10.24.34.xxx>",
                "GitServerDnsName": "<enterprise.git.com>",
                "GitServerIpV4List": [
                    "<xxx.xxx.xxx.xxx>",
                    "<xxx.xxx.xxx.xxx>"
                ]
            },
            {
                "DnsServerIpV4": "<10.24.34.xxx>",
                "GitServerDnsName": "<git.example.com>",
                "GitServerIpV4List": [
                    "<xxx.xxx.xxx.xxx>",
                    "<xxx.xxx.xxx.xxx>"
                ]
            }
        ]
    }
]
```

2. Simpan file konfigurasi Anda sebagai `configuration.json`.
3. Unggah file konfigurasi ke lokasi penyimpanan Amazon S3 yang ditunjuk dalam folder bernama `life-cycle-configuration`. Misalnya, jika lokasi S3 default Anda adalah `s3://DOC-EXAMPLE-BUCKET/notebooks`, file konfigurasi Anda harus berlokasi di `s3://DOC-EXAMPLE-BUCKET/notebooks/`**`life-cycle-configuration/configuration.json`**.

> **Important**
>
> Kami sangat menyarankan agar Anda membatasi akses ke folder `life-cycle-configuration` untuk hanya administrator EMR Notebooks Anda, dan peran layanan untuk EMR Notebook. Anda juga harus mengamankan `configuration.json` terhadap akses yang tidak sah. Untuk instruksi, lihat Mengontrol akses ke bucket dengan kebijakan pengguna atau Praktik Terbaik Keamanan untuk Amazon S3.

Untuk instruksi pengunggahan, lihat Membuat folder dan Pengunggahan objek dalam Panduan Pengguna Amazon Storage Service.

Tambahkan repositori berbasis Git ke Amazon EMR

> **Note**
>
> EMR Notebooks tersedia sebagai EMR Studio Workspaces di konsol baru. Anda masih dapat menggunakan notebook yang ada di konsol lama, tetapi Anda tidak dapat membuat notebook baru di konsol lama. Tombol Create Workspace di konsol baru menggantikan fungsi ini. Untuk mengakses atau membuat Ruang Kerja, pengguna EMR Notebooks memerlukan izin peran IAM tambahan. Untuk informasi selengkapnya, lihat Amazon EMR Notebook adalah Amazon EMR Studio Workspaces di konsol baru dan Apa yang baru di konsol?

Lihat bagian berikut untuk informasi tentang cara menambahkan repositori berbasis Git ke notebook EMR di konsol lama, atau ke EMR Studio Workspace di konsol baru.

New console

Karena EMR Notebooks adalah EMR Studio Workspaces di konsol baru, Anda dapat mengikuti Menautkan repositori berbasis Git ke Workspace EMR Studio instruksi untuk mengaitkan hingga tiga repositori Git dengan Workspace Anda.

Atau, Anda dapat menggunakan JupyterLab Git ekstensi. Pilih ikon Git dari bilah sisi kiri buku catatan Jupyterlab Anda untuk mengakses ekstensi. Untuk informasi tentang ekstensi, lihat repo GitHub jupyterlab-git.

Untuk mengaitkan repositori Git dengan Workspace, administrator Studio Anda harus mengambil langkah-langkah untuk mengonfigurasi Studio agar memungkinkan penautan repositori Git. Untuk informasi selengkapnya, lihat Membuat akses dan izin untuk repositori berbasis Git.

Old console

Untuk menambahkan repositori berbasis Git sebagai sumber daya di akun EMR Amazon Anda dengan konsol lama

1. Buka konsol EMR Amazon lama di https://console.aws.amazon.com/elasticmapreduce.
2. Pilih Repositori Git, lalu pilih Tambahkan repositori.
3. Untuk Nama repositori, masukkan nama yang akan digunakan untuk repositori di Amazon EMR.

   Nama hanya boleh berisi karakter alfanumerik, tanda hubung (-), atau garis bawah (_).
4. Untuk URL repositori Git, masukkan URL untuk repositori. Saat menggunakan CodeCommit repositori, ini adalah URL yang disalin saat Anda memilih Clone URL dan kemudian Clone HTTPS, misalnya,. `https://git-codecommit.us-west-2.amazonaws.com/v1/repos/MyCodeCommitRepoName`
5. Untuk Cabang, masukkan nama cabang.
6. Untuk Kredensial Git, pilih opsi sesuai dengan pedoman berikut. Anda dapat menggunakan nama pengguna Git dan kata sandi atau token akses pribadi (PAT) untuk mengautentikasi ke repositori Anda. EMR Notebooks mengakses kredensial Git Anda menggunakan rahasia yang disimpan di Secrets Manager.

   > **Note**
   >
   > Jika Anda menggunakan GitHub repositori, kami sarankan Anda menggunakan token akses pribadi (PAT) untuk mengautentikasi. Mulai 13 Agustus 2021, tidak GitHub akan lagi menerima kata sandi saat mengautentikasi operasi Git. Untuk informasi selengkapnya, lihat persyaratan otentikasi Token untuk posting operasi Git di GitHub Blog.

Opsi	Deskripsi
Gunakan AWS rahasia yang ada	Pilih opsi ini jika Anda telah menyimpan kredensial Anda sebagai secret di Secrets Manager, lalu pilih nama secret dari daftar. Jika Anda memilih rahasia yang terkait dengan nama pengguna Git dan kata sandi, rahasia harus dalam format `{"gitUsername": " MyUserName ", "gitPassword": " MyPassword "}`.

Opsi	Deskripsi
Buat rahasia baru	Pilih opsi ini untuk mengasosiasikan kredensial Git yang ada dengan rahasia baru yang Anda buat di Secrets Manager. Lakukan salah satu dari berikut ini berdasarkan kredensial Git yang Anda gunakan untuk repositori. Jika Anda menggunakan nama pengguna Git dan kata sandi untuk mengakses repositori, pilih Nama pengguna dan kata sandi, masukkan Nama secret untuk digunakan di Secrets Manager, kemudian masukkan Nama pengguna dan Kata Sandi untuk dikaitkan dengan secret. –ATAU– Jika Anda menggunakan token akses pribadi untuk mengakses repositori, pilih Token akses pribadi (PAT), masukkan Nama secret untuk digunakan di Secrets Manager, kemudian masukkan Token akses pribadi. Untuk informasi selengkapnya, lihat Membuat token akses pribadi untuk baris perintah GitHub dan Token akses pribadi untuk Bitbucket. CodeCommit repositori tidak mendukung opsi ini.
Gunakan repositori publik tanpa kredensyal	Pilih opsi ini untuk mengakses repositori publik.

7. Pilih Menambahkan repositori.

Memperbarui atau menghapus repositori berbasis Git

> **Note**
>
> EMR Notebooks tersedia sebagai EMR Studio Workspaces di konsol baru. Anda masih dapat menggunakan notebook yang ada di konsol lama, tetapi Anda tidak dapat membuat notebook baru di konsol lama. Tombol Create Workspace di konsol baru menggantikan fungsi ini. Untuk mengakses atau membuat Ruang Kerja, pengguna EMR Notebooks memerlukan izin peran IAM tambahan. Untuk informasi selengkapnya, lihat Amazon EMR Notebook adalah Amazon EMR Studio Workspaces di konsol baru dan Apa yang baru di konsol?

Lihat bagian berikut untuk informasi tentang cara menghapus repositori berbasis Git dari notebook EMR di konsol lama, atau dari EMR Studio Workspace di konsol baru.

New console

Karena EMR Notebooks adalah EMR Studio Workspaces di konsol baru, Anda dapat Menautkan repositori berbasis Git ke Workspace EMR Studio merujuk ke informasi selengkapnya tentang bekerja dengan repositori Git di Workspace Anda. Tetapi saat ini, Anda tidak dapat menghapus repositori Git dari Workspaces.

Old console

Untuk memperbarui repositori berbasis Git di konsol lama

1. Pada halaman Repositori Git, pilih repositori yang ingin Anda perbarui.
2. Pada halaman repositori, pilih Edit repositori.
3. Memperbarui Kredensial Git pada halaman repositori.

Untuk menghapus repositori Git di konsol lama

1. Pada halaman Repositori Git, pilih repositori yang ingin Anda hapus.
2. Pada halaman repositori, pilih semua notebook yang saat ini terkait dengan repositori. Pilih Hapus tautan notebook.
3. Pada halaman repositori, pilih Hapus.

> **Note**
>
> Untuk menghapus repositori Git lokal dari Amazon EMR, Anda harus terlebih dahulu menghapus tautan notebook apa pun dari repositori ini. Untuk informasi selengkapnya, lihat Tautkan atau hapus tautan repositori berbasis Git. Menghapus repositori Git tidak akan menghapus rahasia yang dibuat untuk repositori. Anda dapat menghapus rahasia dalam AWS Secrets Manager.

Tautkan atau hapus tautan repositori berbasis Git

> **Note**
>
> EMR Notebooks tersedia sebagai EMR Studio Workspaces di konsol baru. Anda masih dapat menggunakan notebook yang ada di konsol lama, tetapi Anda tidak dapat membuat notebook baru di konsol lama. Tombol Create Workspace di konsol baru menggantikan fungsi ini. Untuk mengakses atau membuat Ruang Kerja, pengguna EMR Notebooks memerlukan izin peran IAM tambahan. Untuk informasi selengkapnya, lihat Amazon EMR Notebook adalah Amazon EMR Studio Workspaces di konsol baru dan Apa yang baru di konsol?

Gunakan langkah-langkah berikut untuk menautkan atau memutuskan tautan repositori berbasis GIS ke notebook EMR di konsol lama, atau ke EMR Studio Workspace di konsol baru.

New console

Karena EMR Notebooks adalah EMR Studio Workspaces di konsol baru, Anda dapat Menautkan repositori berbasis Git ke Workspace EMR Studio merujuk ke informasi selengkapnya tentang bekerja dengan repositori Git di Workspace Anda. Tetapi saat ini, Anda tidak dapat menghapus repositori Git dari Workspaces.

Old console

Untuk menautkan repositori berbasis Git ke EMR notebook

Repositori dapat dihubungkan ke notebook setelah notebook Siap.

1. Dari daftar Notebook, pilih notebook yang ingin Anda perbarui.
2. Di bagian Repositori Git dari halaman Notebook, pilih Tautkan repositori baru.

3. Dalam daftar repositori dari jendela Tautkan repositori Git ke notebook, pilih satu atau lebih repositori yang ingin Anda tautkan ke notebook Anda, kemudian pilih Tautkan repositori.

Atau

1. Pada halaman Repositori Git, pilih repositori yang ingin Anda tautkan ke notebook.
2. Dalam daftar EMR notebook, pilih Tautkan notebook baru untuk menautkan repositori ini ke notebook yang sudah ada.

Untuk menghapus tautan repositori berbasis Git dari EMR notebook

1. Dari daftar Notebook, pilih notebook yang ingin Anda perbarui.
2. Dalam daftar Repositori Git, pilih repositori yang ingin Anda hapus tautannya dari notebook Anda, kemudian pilih Hapus tautan repositori.

Atau

1. Pada halaman Repositori Git, pilih repositori yang ingin Anda perbarui.
2. Dalam daftar EMR notebooks, pilih notebook yang ingin Anda hapus tautannya dari repositori, lalu pilih Hapus tautan notebook.

> **Note**
>
> Menautkan repositori Git ke notebook mengkloning repositori jarak jauh ke notebook Jupyter lokal Anda. Memutuskan tautan repositori Git dari notebook hanya memutus notebook dari repositori jarak jauh tetapi tidak menghapus repositori Git lokal.

Memahami status repositori

Sebuah repositori Git mungkin memiliki salah satu status berikut dalam daftar repositori. Untuk informasi lebih lanjut tentang menautkan EMR notebook dengan repositori Git, lihat Tautkan atau hapus tautan repositori berbasis Git.

Status	Arti
Menautkan	Repositori Git sedang ditautkan dengan notebook. Sementara repositori Menautkan, Anda tidak dapat menghentikan notebook.
Ditautkan	Repositori Git ditautkan dengan notebook. Sementara repositori memiliki status Ditautkan, terhubung ke repositori jarak jauh.
Tautan Gagal	Repositori Git gagal ditautkan dengan notebook. Anda dapat mencoba lagi untuk menautkannya.
Menghapus tautan	Repositori Git sedang dihapus tautannya dari notebook. Sementara repositori Menghapus tautan, Anda tidak dapat menghentikan notebook. Menghapus tautan repositori Git dari notebook hanya memutus sambungan dari repositori jarak jauh tetapi tidak menghapus kode apa pun dari notebook.
Hapus Tautan Gagal	Repositori Git gagal dihapus tautannya dari notebook. Anda dapat mencoba lagi untuk menghapus tautan.

Buat Notebook baru dengan repositori Git terkait

> **Note**
>
> EMR Notebooks tersedia sebagai EMR Studio Workspaces di konsol baru. Anda masih dapat menggunakan notebook yang ada di konsol lama, tetapi Anda tidak dapat membuat notebook baru di konsol lama. Tombol Create Workspace di konsol baru menggantikan fungsi ini. Untuk mengakses atau membuat Ruang Kerja, pengguna EMR Notebooks memerlukan izin peran IAM tambahan. Untuk informasi selengkapnya, lihat Amazon EMR Notebook adalah Amazon EMR Studio Workspaces di konsol baru dan Apa yang baru di konsol?

Untuk membuat buku catatan dan mengaitkannya dengan repositori Git di konsol EMR Amazon lama

1. Ikuti petunjuk di Membuat Notebook.
2. Untuk Grup keamanan, pilih Gunakan grup keamanan Anda sendiri.

> **Note**
>
> Grup keamanan untuk notebook Anda harus menyertakan aturan keluar untuk mengizinkan notebook mengarahkan lalu lintas ke internet dari klaster. Kami menyarankan agar Anda membuat grup keamanan Anda sendiri. Untuk informasi lebih lanjut, lihat Menentukan grup keamanan EC2 untuk EMR Notebooks.

3. Untuk Repositori Git, Pilih repositori untuk mengasosiasikan dengan notebook.
    1. Pilih repositori yang disimpan sebagai sumber daya di akun Anda, lalu pilih Simpan.
    2. Untuk menambahkan repositori baru sebagai sumber daya di akun Anda, pilih Tambahkan repositori baru. Lengkapi alur kerja Menambahkan repositori di jendela baru.

Gunakan repositori Git di Notebook

> **Note**
>
> EMR Notebooks tersedia sebagai EMR Studio Workspaces di konsol baru. Anda masih dapat menggunakan notebook yang ada di konsol lama, tetapi Anda tidak dapat membuat notebook baru di konsol lama. Tombol Create Workspace di konsol baru menggantikan fungsi ini. Untuk mengakses atau membuat Ruang Kerja, pengguna EMR Notebooks memerlukan izin peran IAM tambahan. Untuk informasi selengkapnya, lihat Amazon EMR Notebook adalah Amazon EMR Studio Workspaces di konsol baru dan Apa yang baru di konsol?

Anda dapat memilih untuk Buka di JupyterLab atau Buka di Jupyter saat Anda membuka buku catatan.

Jika Anda memilih untuk membuka notebook di Jupyter, daftar file dan folder yang dapat diperluas di dalam notebook akan ditampilkan. Anda dapat secara manual menjalankan perintah Git seperti berikut dalam sel notebook.

```
!git pull origin primary
```

Untuk membuka salah satu repositori tambahan, navigasikan ke folder lain.

Jika Anda memilih untuk membuka notebook dengan JupyterLab antarmuka, Anda dapat menggunakan ekstensi JupyterLab Git yang sudah diinstal sebelumnya. Untuk informasi tentang ekstensi, lihat jupyterlab-git.

Merencanakan dan mengonfigurasi klaster

Bagian ini menjelaskan pilihan konfigurasi dan petunjuk untuk merencanakan, mengonfigurasi, dan meluncurkan klaster menggunakan Amazon EMR. Sebelum Anda meluncurkan klaster, silakan buat pilihan tentang sistem Anda berdasarkan data yang Anda proses dan kebutuhan Anda untuk biaya, kecepatan, kapasitas, ketersediaan, keamanan, dan pengelolaan. Pilihan Anda mencakup:

- Di wilayah mana klaster akan dijalankan, di mana dan bagaimana untuk menyimpan data, dan bagaimana hasil outputnya. Lihat Mengkonfigurasi lokasi klaster dan penyimpanan data.
- Apakah Anda menjalankan klaster Amazon EMR di Outposts atau Local Zones. Lihat Klaster EMR pada AWS Outposts atau Klaster EMR di AWS Local Zones.
- Apakah klaster berjalan lama atau sementara, dan perangkat lunak apa yang berjalan. Lihat Mengkonfigurasi cluster untuk melanjutkan atau mengakhiri setelah eksekusi langkah dan Konfigurasi perangkat lunak klaster.
- Apakah sebuah cluster memiliki satu node primer atau tiga node primer. Lihat Rencanakan dan konfigurasikan node primer.
- Opsi perangkat keras dan jaringan yang mengoptimalkan biaya, kinerja, dan ketersediaan untuk aplikasi Anda. Lihat Konfigurasi perangkat keras dan jaringan klaster.
- Cara mengatur klaster sehingga Anda dapat mengelolanya dengan lebih mudah, dan memantau aktivitas, kinerja, dan kesehatan. Lihat Konfigurasi pencatatan log dan debugging klaster dan Klaster tag.
- Cara mengautentikasi dan mengotorisasi akses ke klaster sumber daya, dan cara mengenkripsi data. Lihat Keamanan di Amazon EMR.
- Cara mengintegrasikan dengan perangkat lunak dan layanan lainnya. Lihat Driver dan integrasi aplikasi pihak ketiga.

Luncurkan cluster dengan cepat

> **Note**
>
> Kami telah mendesain ulang konsol EMR Amazon agar lebih mudah digunakan. Lihat Apa yang baru dengan konsol? untuk mempelajari tentang perbedaan antara pengalaman konsol lama dan baru.

New console

Untuk meluncurkan cluster dengan konsol baru dengan cepat

1. Masuk keAWS Management Console, dan buka konsol EMR Amazon di https://console.aws.amazon.com/emr/clusters.
2. Di bawah EMR pada EC2 di panel navigasi kiri, pilih Clusters, lalu pilih Create cluster.
3. Pada halaman Buat Cluster, masukkan atau pilih nilai untuk bidang yang disediakan. Panel ringkasan persisten menampilkan tampilan real-time dari opsi cluster yang Anda pilih saat ini. Pilih judul di panel ringkasan untuk menavigasi ke bagian yang sesuai dan melakukan penyesuaian. Anda harus menyelesaikan semua konfigurasi yang diperlukan sebelum Anda dapat memilih Buat cluster.
4. Pilih Buat cluster untuk menerima konfigurasi seperti yang ditunjukkan.
5. Halaman detail cluster terbuka. Temukan Status cluster di sebelah nama cluster. Status harus berubah dari Memulai ke berjalan ke Menunggu selama proses pembuatan klaster. Anda mungkin perlu memilih ikon penyegaran di kanan atas atau menyegarkan browser Anda untuk menerima pembaruan.

   Ketika status berubah ke Menunggu, klaster Anda siap, berjalan, dan siap menerima langkah-langkah dan koneksi SSH.

Old console

Gunakan halaman Create Cluster - Quick Options di konsol EMR Amazon lama untuk membuat cluster dengan cepat untuk tugas-tugas sederhana atau untuk tujuan evaluasi atau pengujian. Opsi Cepat menggunakan nilai default untuk opsi konfigurasi seperti perangkat lunak klaster, jaringan, dan keamanan.

Untuk meluncurkan cluster dengan Opsi Cepat dengan konsol lama

1. Arahkan ke konsol EMR Amazon baru dan pilih Beralih ke konsol lama dari navigasi samping. Untuk informasi selengkapnya tentang apa yang diharapkan saat beralih ke konsol lama, lihat Menggunakan konsol lama.
2. Pilih Klaster, lalu pilih Buat klaster untuk membuka halaman Opsi Cepat.
3. Di halaman Buat Klaster - Opsi Cepat, masukkan atau pilih nilai untuk bidang yang disediakan.
4. Pilih Buat klaster untuk meluncurkan klaster dan membuka halaman status klaster.

5. Di halaman status klaster, cari Status klaster di sebelah nama klaster. Status harus berubah dari Memulai ke berjalan ke Menunggu selama proses pembuatan klaster. Anda mungkin harus memilih ikon segarkan di sebelah kanan atau segarkan peramban Anda untuk menerima pembaruan.

   Ketika status berubah ke Menunggu, klaster Anda siap, berjalan, dan siap menerima langkah-langkah dan koneksi SSH.

Mengkonfigurasi lokasi klaster dan penyimpanan data

Bagian ini menjelaskan cara mengonfigurasi wilayah untuk klaster, sistem file yang berbeda tersedia saat Anda menggunakan Amazon EMR dan cara menggunakannya. Hal ini juga mencakup cara mempersiapkan atau mengunggah data ke Amazon EMR jika perlu, serta cara mempersiapkan lokasi output untuk berkas log dan file data output yang Anda konfigurasikan.

Topik

Pilih Wilayah AWS

Amazon Web Services berjalan di server di pusat data di seluruh dunia. Pusat data diatur oleh Wilayah geografis. Saat meluncurkan kluster EMR Amazon, Anda harus menentukan Wilayah. Anda dapat memilih Wilayah untuk mengurangi latensi, meminimalkan biaya, atau memenuhi persyaratan peraturan. Untuk daftar Wilayah dan titik akhir yang didukung oleh Amazon EMR, lihat Wilayah dan titik akhir di. Referensi Umum Amazon Web Services

Untuk performa terbaik, Anda harus meluncurkan cluster di Wilayah yang sama dengan data Anda. Misalnya, jika bucket Amazon S3 yang menyimpan data input Anda berada di Wilayah AS Barat (Oregon), Anda harus meluncurkan cluster Anda di Wilayah AS Barat (Oregon) untuk menghindari biaya transfer data lintas wilayah. Jika Anda menggunakan bucket Amazon S3 untuk menerima output cluster, Anda juga ingin membuatnya di Wilayah AS Barat (Oregon).

Jika Anda berencana untuk mengaitkan key pair Amazon EC2 dengan cluster (diperlukan untuk menggunakan SSH untuk masuk ke master node), key pair harus dibuat di Region yang sama

dengan cluster. Demikian pula, grup keamanan yang dibuat Amazon EMR untuk mengelola cluster dibuat di Wilayah yang sama dengan cluster.

Jika Anda mendaftar Akun AWS pada atau setelah 17 Mei 2017, Wilayah default saat Anda mengakses sumber daya dari AWS Management Console adalah US East (Ohio) (us-timur-2); untuk akun yang lebih lama, Wilayah default adalah US West (Oregon) (us-barat-2) atau AS Timur (Virginia N) (us-timur-1). Untuk informasi selengkapnya, lihat Wilayah dan Titik Akhir.

Beberapa AWS fitur hanya tersedia di Wilayah terbatas. Misalnya, instans Cluster Compute hanya tersedia di Wilayah AS Timur (Virginia N.), dan Wilayah Asia Pasifik (Sydney) hanya mendukung Hadoop 1.0.3 dan yang lebih baru. Saat memilih Wilayah, periksa apakah itu mendukung fitur yang ingin Anda gunakan.

Untuk kinerja terbaik, gunakan Wilayah yang sama untuk semua sumber AWS daya Anda yang akan digunakan dengan cluster. Tabel berikut memetakan nama Wilayah antar layanan. Untuk daftar Wilayah EMR Amazon, lihat Wilayah AWSdan titik akhir di. Referensi Umum Amazon Web Services

Pilih Wilayah dengan konsol

Wilayah default Anda ditampilkan di sebelah kiri informasi akun Anda di bilah navigasi. Untuk mengganti Wilayah di konsol baru dan lama, pilih menu tarik-turun Wilayah dan pilih opsi baru.

Tentukan Wilayah dengan AWS CLI

Tentukan Wilayah default dalam AWS CLI menggunakan aws configure perintah atau variabel `AWS_DEFAULT_REGION` lingkungan. Untuk informasi selengkapnya, lihat Mengonfigurasi AWS Wilayah di AWS Command Line InterfacePanduan Pengguna.

Pilih Wilayah dengan SDK atau API

Untuk memilih Wilayah menggunakan SDK, konfigurasikan aplikasi Anda untuk menggunakan titik akhir Wilayah tersebut. Jika Anda membuat aplikasi klien menggunakan SDK AWS, Anda dapat mengubah titik akhir klien dengan memanggil `setEndpoint`, seperti yang ditunjukkan dalam contoh berikut:

```
client.setEndpoint("elasticmapreduce.us-west-2.amazonaws.com");
```

Setelah aplikasi Anda menetapkan Region dengan menyetel titik akhir, Anda dapat mengatur Availability Zone untuk instance EC2 klaster Anda. Availability Zones adalah lokasi geografis

berbeda yang dirancang untuk diisolasi dari kegagalan di Availability Zone lainnya dan menyediakan konektivitas jaringan latensi rendah yang murah ke Availability Zone lainnya di Wilayah yang sama. Sebuah Wilayah berisi satu atau beberapa Availability Zone. Untuk mengoptimalkan kinerja dan mengurangi latensi, semua sumber daya harus terletak di Availability Zone yang sama sebagai klaster yang menggunakan mereka.

Bekerja dengan sistem penyimpanan dan file

Amazon EMR dan Hadoop menyediakan berbagai sistem file yang dapat Anda gunakan saat memproses langkah-langkah klaster. Anda menentukan sistem file yang akan digunakan oleh prefiks URI yang digunakan untuk mengakses data. Misalnya, `s3://DOC-EXAMPLE-BUCKET1/path` referensi bucket Amazon S3 menggunakan EMRFS. Tabel berikut mencantumkan sistem file yang tersedia, dengan rekomendasi tentang kapan sebaiknya masing-masing digunakan.

Amazon EMR dan Hadoop biasanya menggunakan dua atau lebih dari sistem file berikut saat memproses klaster. HDFS dan EMRFS adalah dua sistem file utama yang digunakan dengan Amazon EMR.

> **Important**
>
> Dimulai dengan rilis Amazon EMR 5.22.0, Amazon EMR AWS menggunakan Signature Version 4 secara eksklusif untuk mengautentikasi permintaan ke Amazon S3. Rilis Amazon EMR sebelumnya menggunakan AWS Signature Version 2 dalam beberapa kasus, kecuali catatan rilis menunjukkan bahwa Signature Version 4 digunakan secara eksklusif. Untuk informasi selengkapnya, lihat Mengautentikasi Permintaan (AWS Versi Tanda Tangan 4) dan Mengautentikasi permintaan (AWSVersi Tanda Tangan 2) di Panduan Developer Amazon Simple Storage Service.

Sistem file	Prefiks	Deskripsi
HDFS	`hdfs://` (atau tanpa prefiks)	HDFS adalah sistem file terdistribusi, dapat diskalakan, dan portabel untuk Hadoop. Keuntungan dari HDFS adalah kesadaran data antara simpul klaster Hadoop yang mengelola klaster dan simpul klaster Hadoop yang mengelola langkah-langkah individu. Untuk informasi selengkapnya, lihat Dokumentasi Hadoop.

Sistem file	Prefiks	Deskripsi
		HDFS digunakan oleh simpul master dan inti. Salah satu keuntungannya adalah cepat; kerugiannya adalah penyimpanan sementara yang direklamasi ketika klaster berakhir. Ini paling baik digunakan untuk melakukan cache hasil yang dibuat oleh langkah-l angkah alur kerja menengah.
EMRFS	`s3://`	EMRFS merupakan implementasi dari sistem file Hadoop yang digunakan untuk membaca dan menulis file reguler dari Amazon EMR langsung ke Amazon S3. EMRFS memberikan kemudahan menyimpan data persisten di Amazon S3 untuk digunakan dengan Hadoop sambil juga menyediakan fitur seperti enkripsi sisi server Amazon S3, konsistensi, dan konsistensi daftar. read-after-write **Note** Sebelumnya, Amazon EMR menggunak an sistem file `s3n` dan `s3a`. Sementara keduanya masih bekerja, kami sarankan Anda menggunakan Skema URI `s3` untuk kinerja, keamanan, dan keandalan terbaik.

Sistem file	Prefiks	Deskripsi
Sistem file lokal		Sistem file lokal mengacu pada disk yang terhubung secara lokal. Ketika klaster Hadoop dibuat, setiap simpul dibuat dari instans EC2 yang datang dengan blok yang telah dikonfigurasikan dari penyimpanan disk dipasang sebelumnya disebut penyimpanan instans. Data pada volume penyimpanan instans hanya bertahan selama masa pakai instans EC2. Volume penyimpanan instans cocok untuk menyimpan data sementara yang terus berubah, seperti buffer, cache, data scratch, dan konten sementara lainnya. Untuk informasi selengkapnya, lihat Penyimpanan instans Amazon EC2. Sistem file lokal digunakan oleh HDFS, tetapi Python juga berjalan dari sistem file lokal dan Anda dapat memilih untuk menyimpan file aplikasi tambahan pada volume penyimpanan instance.
Sistem file blok Amazon S3 (Legasi)	`s3bfs://`	Sistem file blok Amazon S3 adalah sistem penyimpan an file legasi. Kami sangat mencegah penggunaan sistem ini. **Important** Kami sarankan Anda tidak menggunak an sistem file ini karena dapat memicu kondisi balapan yang mungkin menyebabkan kegagalan klaster Anda. Namun, ini mungkin diperlukan oleh aplikasi warisan.

Mengakses sistem file

Anda menentukan sistem file mana yang akan digunakan oleh prefiks pengidentifikasi sumberdaya seragam (URI) yang digunakan untuk mengakses data. Prosedur berikut menggambarkan cara mereferensikan beberapa jenis sistem file yang berbeda.

Untuk mengakses HDFS lokal

- Tentukan `hdfs:///` prefiks dalam URI. Amazon EMR menyelesaikan jalur yang tidak menentukan prefiks dalam URI ke HDFS lokal. Sebagai contoh, kedua URI berikut akan menyelesaikan di lokasi yang sama dalam HDFS.

```
hdfs:///path-to-data

/path-to-data
```

Untuk mengakses HDFS secara jarak jauh

- Sertakan alamat IP simpul master di URI, sebagaimana yang ditunjukkan dalam contoh berikut.

```
hdfs://master-ip-address/path-to-data

master-ip-address/path-to-data
```

Untuk mengakses Amazon S3

- Gunakan `s3://` prefiks.

```
s3://bucket-name/path-to-file-in-bucket
```

Untuk mengakses sistem file blok Amazon S3

- Gunakan hanya untuk aplikasi warisan yang membutuhkan sistem file blok Amazon S3. Untuk mengakses atau menyimpan data dengan sistem file ini, gunakan `s3bfs://` prefiks dalam URI.

  Sistem file blok Amazon S3 adalah sistem file warisan yang digunakan untuk mendukung pengunggahan ke Amazon S3 yang berukuran lebih besar dari 5 GB. Dengan fungsionalitas unggahan multipart, Amazon EMR menyediakan melalui AWS Java SDK, Anda dapat mengunggah file yang berukuran hingga 5 TB ke sistem file asli Amazon S3, dan sistem file blok Amazon S3 tidak lagi digunakan.

  > ⚠ Warning
  >
  > Karena sistem file warisan ini dapat membuat kondisi balapan yang dapat merusak sistem file, Anda harus menghindari format ini dan menggunakan EMRFS.

  ```
  s3bfs://bucket-name/path-to-file-in-bucket
  ```

Mempersiapkan data input

Sebagian besar klaster memuat data input kemudian memproses data tersebut. Untuk memuat data, data harus berada di lokasi yang mana dapat diakses oleh klaster dan dalam format yang dapat diproses oleh klaster. Skenario yang paling umum adalah mengunggah data input ke Amazon S3. Amazon EMR menyediakan alat untuk klaster Anda yang mana digunakan mengimpor atau membaca data dari Amazon S3.

Format input default dalam Hadoop adalah file teks, meskipun Anda dapat menyesuaikan Hadoop dan menggunakan alat untuk mengimpor data yang disimpan dalam format lain.

Topik

- Jenis input yang dapat diterima Amazon EMR
- Cara memasukkan data ke Amazon EMR

Jenis input yang dapat diterima Amazon EMR

Format input default untuk klaster adalah file teks dengan setiap baris dipisahkan oleh karakter baris baru (\n), yang merupakan format input yang paling sering digunakan.

Jika data input Anda dalam format selain file teks default, Anda bisa menggunakan antarmuka Hadoop `InputFormat` untuk menentukan jenis input lainnya. Anda bahkan dapat membuat subkelas dari kelas `FileInputFormat` untuk menangani jenis data khusus. Untuk informasi lebih lanjut, lihat http://hadoop.apache.org/docs/current/api/org/apache/hadoop/mapred/ InputFormat .html.

Jika Anda menggunakan Hive, Anda dapat menggunakan serializer/deserializer (SerDe) untuk membaca data dari format tertentu ke HDFS. Untuk informasi lebih lanjut, lihat https://cwiki.apache.org/confluence/display/Hive/ SerDe.

Cara memasukkan data ke Amazon EMR

Amazon EMR menyediakan beberapa cara untuk memasukkan data ke dalam klaster. Cara paling umum adalah dengan mengunggah data ke Amazon S3 dan menggunakan fitur bawaan Amazon EMR untuk mengunggah data ke klaster Anda. Anda juga dapat menggunakan DistributedCache fitur Hadoop untuk mentransfer file dari sistem file terdistribusi ke sistem file lokal. Implementasi Hive yang disediakan oleh Amazon EMR (Hive versi 0.7.1.1 dan yang lebih baru) mencakup fungsionalitas yang dapat Anda gunakan untuk mengimpor dan mengekspor data antara DynamoDB dan klaster Amazon EMR. Jika Anda memiliki data lokal dalam jumlah besar untuk diproses, Anda mungkin merasa jika layanan AWS Direct Connect ini berguna.

Topik

- Mengunggah data ke Amazon S3
- Unggah data dengan AWS DataSync
- Impor file dengan cache terdistribusi
- Cara memproses file terkompresi
- Mengimpor data DynamoDB ke Hive
- Connect ke data dengan AWS Direct Connect
- Unggah data dalam jumlah besar dengan AWS Snowball

Mengunggah data ke Amazon S3

Untuk informasi tentang cara mengunggah objek ke Amazon S3, lihat Menambahkan objek ke bucket di Panduan Pengguna Layanan Penyimpanan Sederhana Amazon. Untuk informasi selengkapnya tentang menggunakan Amazon S3 dengan Hadoop, lihat http://wiki.apache.org/hadoop/AmazonS3.

Topik

- Buat dan konfigurasikan bucket Amazon S3
- Mengonfigurasi unggahan multipart untuk Amazon S3
- Praktik terbaik
- Unggah data ke Amazon S3 Express One Zone

Buat dan konfigurasikan bucket Amazon S3

Amazon EMR menggunakan AWS SDK for Java dengan Amazon S3 untuk menyimpan data input, berkas log, dan data output. Amazon S3 mengacu pada lokasi penyimpanan ini sebagai bucket. Bucket memiliki pembatasan dan batasan tertentu agar sesuai dengan persyaratan Amazon S3 dan DNS. Untuk informasi selengkapnya, lihat Pembatasan dan batasan Bucket di Panduan Pengguna Layanan Penyimpanan Sederhana Amazon.

Bagian ini menunjukkan cara menggunakan Amazon S3 AWS Management Console Untuk membuat lalu menetapkan izin untuk bucket Amazon S3. Anda juga dapat membuat dan mengatur izin untuk bucket Amazon S3 menggunakan API Amazon S3 atau AWS CLI. Anda juga bisa menggunakan curl bersama dengan modifikasi untuk meneruskan parameter autentikasi yang sesuai bagi Amazon S3.

Lihat sumber daya berikut:

- Untuk membuat bucket menggunakan konsol, lihat Membuat bucket di Panduan Pengguna Amazon S3.
- Untuk membuat dan bekerja dengan bucket menggunakanAWS CLI, lihat Menggunakan perintah S3 tingkat tinggi dengan AWS Command Line Interface di Panduan Pengguna Amazon S3.
- Untuk membuat bucket menggunakan SDK, lihat Contoh membuat bucket di Panduan Pengguna Layanan Penyimpanan Sederhana Amazon.
- Untuk bekerja dengan bucket menggunakan curl, lihat alat autentikasi Amazon S3 untuk curl.
- Untuk informasi selengkapnya tentang menentukan bucket khusus Wilayah, lihat Mengakses bucket di Panduan Pengguna Layanan Penyimpanan Sederhana Amazon.

- Untuk bekerja dengan bucket menggunakan Titik Akses Amazon S3, lihat Menggunakan alias gaya ember untuk titik akses Anda di Panduan Pengguna Amazon S3. Anda dapat dengan mudah menggunakan Titik Akses Amazon S3 dengan Alias Titik Akses Amazon S3 alih-alih nama bucket Amazon S3. Anda dapat menggunakan Alias Titik Akses Amazon S3 untuk aplikasi yang ada dan yang baru, termasuk Spark, Hive, Presto, dan lainnya.

> **Note**
>
> Jika Anda mengaktifkan pencatatan log untuk bucket, ini hanya mengaktifkan log akses bucket, bukan log klaster Amazon EMR.

Selama pembuatan bucket atau setelahnya, Anda dapat mengatur izin yang sesuai untuk mengakses bucket, bergantung pada aplikasi Anda. Biasanya, Anda memberi diri Anda (pemilik) akses baca dan tulis dan memberi akses baca untuk pengguna yang diautentikasi.

Bucket Amazon S3 yang diperlukan harus ada sebelum Anda dapat membuat klaster. Anda harus mengunggah skrip atau data yang diperlukan yang dimaksud dalam klaster ke Amazon S3. Tabel berikut menjelaskan contoh data, skrip, dan lokasi berkas log.

Mengonfigurasi unggahan multipart untuk Amazon S3

Amazon EMR mendukung unggahan multipart Amazon S3 melalui AWS SDK for Java. Unggahan multipart memungkinkan Anda mengunggah satu objek ke dalam beberapa bagian. Anda dapat mengunggah bagian-bagian objek tersebut secara independen dan dengan urutan apa pun. Jika ada transmisi bagian mana pun yang gagal, Anda dapat mentransmisikan ulang bagian tersebut tanpa memengaruhi bagian lainnya. Setelah semua bagian objek Anda diunggah, Amazon S3 merakit bagian-bagian tersebut dan menciptakan objek.

Untuk informasi selengkapnya, lihat Ikhtisar unggahan multibagian di Panduan Pengguna Layanan Penyimpanan Sederhana Amazon.

Selain itu, Amazon EMR menawarkan properti yang memungkinkan Anda mengontrol pembersihan bagian unggahan multipart yang gagal dengan lebih tepat.

Tabel berikut menjelaskan properti konfigurasi Amazon EMR untuk unggahan multipart. Anda dapat mengonfigurasi `core-site` menggunakan klasifikasi konfigurasi. Untuk informasi selengkapnya, lihat Konfigurasi aplikasi di Panduan Rilis Amazon EMR.

Nama parameter konfigurasi	Nilai default	Deskripsi
`fs.s3n.multipart.uploads.enabled`	`true`	Jenis Boolean yang menunjukkan apakah akan mengaktifkan unggahan multipart. Saat tampilan konsisten EMRFS diaktifkan, unggahan multibagian diaktifkan secara default dan menyetel nilai ini diabaikan. `false`
`fs.s3n.multipart.uploads.split.size`	`134217728`	Menentukan ukuran maksimum dari bagian, dalam byte, sebelum EMRFS memulai pengunggahan bagian baru saat unggahan multipart diaktifkan. Nilai minimumnya adalah `5242880` (5 MB). Jika nilai yang lebih rendah ditentukan, `5242880` digunakan. Maksimumnya adalah `5368709120` (5 GB). Jika nilai yang lebih besar ditentukan, `5368709120` digunakan. Jika enkripsi di sisi klien EMRFS dinonaktifkan dan Amazon S3 Optimized Committer juga dinonaktifkan, nilai ini juga mengontrol ukuran maksimum yang dapat dikembangkan file data hingga EMRFS menggunakan unggahan multipart alih-alih permintaan `PutObject` untuk mengunggah file. Untuk informasi selengkapnya, lihat
`fs.s3n.ssl.enabled`	`true`	Jenis Boolean yang menunjukkan apakah akan menggunakan http atau https.
`fs.s3.buckets.create.enabled`	`false`	Jenis Boolean yang menunjukkan apakah bucket harus dibuat jika tidak ada. Mengatur ke `false` menyebabkan pengecualian pada `CreateBucket` operasi.

Nama parameter konfigurasi	Nilai default	Deskripsi
`fs.s3.multipart.clean.enabled`	`false`	Jenis Boolean yang menunjukkan apakah akan mengaktifkan pembersihan berkala latar belakang dari unggahan multipart yang tidak lengkap.
`fs.s3.multipart.clean.age.threshold`	`604800`	Jenis panjang yang menentukan usia minimum dari unggahan multipart, dalam hitungan detik, sebelum dipertimbangkan untuk dibersihkan. Default adalah satu minggu.
`fs.s3.multipart.clean.jitter.max`	`10000`	Jenis integer yang menentukan jumlah maksimum penundaan jitter acak dalam detik yang ditambahkan ke penundaan tetap 15 menit sebelum menjadwalkan putaran pembersihan berikutnya.

Nonaktifkan unggahan multipart

> **Note**
>
> Kami telah mendesain ulang konsol EMR Amazon agar lebih mudah digunakan. Lihat Apa yang baru dengan konsol? untuk mempelajari tentang perbedaan antara pengalaman konsol lama dan baru.

New console

Untuk menonaktifkan unggahan multipart dengan konsol baru

1. Masuk keAWS Management Console, dan buka konsol EMR Amazon di https://console.aws.amazon.com/emr.
2. Di bawah EMR pada EC2 di panel navigasi kiri, pilih Clusters, lalu pilih Create cluster.

3. Di bawah Pengaturan perangkat lunak, masukkan konfigurasi berikut:`classification=core-site,properties=[fs.s3n.multipart.uploads.enabled=false]`.
4. Pilih opsi lain yang berlaku untuk cluster Anda.
5. Untuk meluncurkan klaster Anda, pilih Buat klaster.

Old console

Untuk menonaktifkan unggahan multipart dengan konsol lama

1. Arahkan ke konsol EMR Amazon baru dan pilih Beralih ke konsol lama dari navigasi samping. Untuk informasi selengkapnya tentang apa yang diharapkan saat beralih ke konsol lama, lihat Menggunakan konsol lama.
2. Pilih Buat klaster, Buka opsi lanjutan.
3. Di bawah Edit Pengaturan Perangkat Lunak, masukkan konfigurasi berikut: `classification=core-site,properties=[fs.s3n.multipart.uploads.enabled=false]`
4. Lanjutkan dengan membuat klaster.

CLI

Untuk menonaktifkan unggahan multipart menggunakan AWS CLI

Prosedur ini menjelaskan cara menonaktifkan unggahan multipart dengan menggunakan file AWS CLI. Untuk menonaktifkan unggahan multipart, ketik perintah `create-cluster` dengan parameter `--bootstrap-actions`.

1. Buat file, `myConfig.json`, dengan konten berikut kemudian simpan di direktori yang sama di mana Anda menjalankan perintah:

```
[
  {
    "Classification": "core-site",
    "Properties": {
      "fs.s3n.multipart.uploads.enabled": "false"
    }
  }
```

```
]
```

2. Ketik perintah berikut dan ganti *MyKey* dengan nama pasangan kunci EC2 anda.

> **Note**
>
> Karakter lanjutan baris Linux (\) disertakan agar mudah dibaca. Karakter ini bisa dihapus atau digunakan dalam perintah Linux. Untuk Windows, hapus atau ganti dengan tanda sisipan (^).

```
aws emr create-cluster --name "Test cluster" \
--release-label emr-7.0.0 --applications Name=Hive Name=Pig \
--use-default-roles --ec2-attributes KeyName=myKey --instance-type m5.xlarge \
--instance-count 3 --configurations file://myConfig.json
```

CLI

Untuk menonaktifkan unggahan multibagian menggunakan API

- Untuk informasi tentang penggunaan unggahan multipart Amazon S3 secara terprogram, lihat Menggunakan SDK for Java untuk upload multipart di AWS Panduan Pengguna Layanan Penyimpanan Sederhana Amazon.

  Untuk informasi lebih lanjut tentang AWS SDK for Java, lihat AWS SDK for Java

Praktik terbaik

Berikut ini adalah rekomendasi untuk menggunakan bucket Amazon S3 dengan klaster EMR.

Aktifkan versioning

Versioning adalah konfigurasi yang direkomendasikan untuk bucket Amazon S3. Dengan mengaktifkan versioning, dapat dipastikan bahwa jika data Anda tidak sengaja dihapus atau ditimpa, data tersebut masih dapat dipulihkan. Untuk informasi selengkapnya, lihat Menggunakan versi di Panduan Pengguna Layanan Penyimpanan Sederhana Amazon.

Bersihkan unggahan multipart yang gagal

Komponen klaster EMR menggunakan unggahan multipart melalui AWS SDK for Java dengan API Amazon S3 untuk menulis berkas log dan data output ke Amazon S3 secara default. Untuk informasi tentang mengubah properti yang terkait dengan konfigurasi ini menggunakan Amazon EMR, lihat Mengonfigurasi unggahan multipart untuk Amazon S3. Terkadang unggahan file besar dapat mengakibatkan unggahan multipart Amazon S3 menjadi tidak lengkap. Jika unggahan multipart tidak berhasil diselesaikan, unggahan multipart yang sedang berlangsung akan terus menempati bucket Anda dan menimbulkan biaya penyimpanan. Kami merekomendasikan opsi berikut untuk menghindari penyimpanan file yang berlebihan:

- Untuk bucket yang Anda gunakan dengan Amazon EMR, gunakan aturan konfigurasi siklus hidup di Amazon S3 untuk menghapus unggahan multipart yang tidak lengkap tiga hari setelah tanggal inisiasi unggahan. Aturan konfigurasi siklus hidup memungkinkan Anda mengontrol kelas penyimpanan dan masa pakai objek. Untuk informasi selengkapnya, lihat Manajemen siklus hidup objek, dan Membatalkan unggahan multipart yang tidak lengkap menggunakan kebijakan siklus hidup bucket.
- Aktifkan fitur pembersihan multipart Amazon EMR dengan mengatur `fs.s3.multipart.clean.enabled` ke `true` dan menyetel parameter pembersihan lainnya. Fitur ini berguna pada volume tinggi, skala besar, dan klaster yang memiliki waktu aktif terbatas. Dalam hal ini, parameter `DaysAfterIntitiation` dari aturan konfigurasi siklus hidup mungkin terlalu panjang, bahkan jika diatur ke minimum, yang menyebabkan lonjakan penyimpanan Amazon S3. Pembersihan multipart Amazon EMR memungkinkan kontrol yang lebih presisi. Untuk informasi selengkapnya, lihat Mengonfigurasi unggahan multipart untuk Amazon S3.

Mengelola penanda versi

Kami menyarankan Anda mengaktifkan aturan konfigurasi siklus hidup di Amazon S3 untuk menghapus delete marker objek kedaluwarsa pada bucket berversi yang Anda gunakan dengan Amazon EMR. Saat menghapus objek dalam bucket berversi, delete marker akan dibuat. Jika semua versi objek sebelumnya kemudian kedaluwarsa, delete marker objek yang kedaluwarsa akan tertinggal di bucket. Meskipun Anda tidak dikenakan biaya untuk menghapus penanda, menghapus penanda yang kedaluwarsa dapat meningkatkan kinerja permintaan LIST. Untuk informasi selengkapnya, lihat Konfigurasi Siklus Hidup untuk bucket dengan pembuatan versi di Panduan Pengguna Layanan Penyimpanan Sederhana Amazon.

Praktik terbaik kinerja

Bergantung pada beban kerja Anda, jenis penggunaan tertentu dari klaster EMR dan aplikasi pada klaster tersebut dapat mengakibatkan jumlah permintaan yang tinggi terhadap bucket. Untuk informasi selengkapnya, lihat Pertimbangan tingkat permintaan dan performa di Panduan Pengguna Layanan Penyimpanan Sederhana Amazon.

Unggah data ke Amazon S3 Express One Zone

Gambaran Umum

Dengan Amazon EMR 6.15.0 dan yang lebih tinggi, Anda dapat menggunakan Amazon EMR dengan Apache Spark bersama dengan kelas penyimpanan Amazon S3 Express One Zone untuk meningkatkan kinerja pada pekerjaan Spark Anda. S3 Express One Zone adalah kelas penyimpanan S3 untuk aplikasi yang sering mengakses data dengan ratusan ribu permintaan per detik. Pada saat rilis, S3 Express One Zone memberikan latensi terendah dan penyimpanan objek cloud kinerja tertinggi di Amazon S3.

Prasyarat

- Izin S3 Express One Zone — Ketika S3 Express One Zone awalnya melakukan tindakan sepertiGET,LIST, atau PUT pada objek S3, kelas penyimpanan memanggil `CreateSession` atas nama Anda. Kebijakan IAM Anda harus mengizinkan `s3express:CreateSession` izin agar S3A konektor dapat menjalankan API. `CreateSession` Untuk contoh kebijakan dengan izin ini, lihatMemulai dengan Amazon S3 Express One Zone.
- S3Akonektor — Untuk mengonfigurasi cluster Spark Anda untuk mengakses data dari bucket Amazon S3 yang menggunakan kelas penyimpanan S3 Express One Zone, Anda harus menggunakan konektor Apache Hadoop. S3A Untuk menggunakan konektor, pastikan semua URI S3 menggunakan skema. `s3a` Jika tidak, Anda dapat mengubah implementasi sistem file yang Anda gunakan untuk `s3` dan skema. `s3n`

Untuk mengubah `s3` skema, tentukan konfigurasi cluster berikut:

```
[
  {
    "Classification": "core-site",
    "Properties": {
      "fs.s3.impl": "org.apache.hadoop.fs.s3a.S3AFileSystem",
      "fs.AbstractFileSystem.s3.impl": "org.apache.hadoop.fs.s3a.S3A"
    }
```

```
  }
]
```

Untuk mengubah `s3n` skema, tentukan konfigurasi cluster berikut:

```
[
  {
    "Classification": "core-site",
    "Properties": {
      "fs.s3n.impl": "org.apache.hadoop.fs.s3a.S3AFileSystem",
      "fs.AbstractFileSystem.s3n.impl": "org.apache.hadoop.fs.s3a.S3A"
    }
  }
]
```

Memulai dengan Amazon S3 Express One Zone

Topik

- Buat kebijakan izin
- Buat dan konfigurasikan cluster Anda
- Ikhtisar konfigurasi

Buat kebijakan izin

Sebelum Anda dapat membuat klaster yang menggunakan Amazon S3 Express One Zone, Anda harus membuat kebijakan IAM untuk melampirkan ke profil instans Amazon EC2 untuk cluster. Kebijakan harus memiliki izin untuk mengakses kelas penyimpanan S3 Express One Zone. Contoh kebijakan berikut menunjukkan cara memberikan izin yang diperlukan. Setelah membuat kebijakan, lampirkan kebijakan ke peran profil instance yang Anda gunakan untuk membuat klaster EMR, seperti yang dijelaskan di bagian iniBuat dan konfigurasikan cluster Anda.

```
{
  "Version":"2012-10-17",
  "Statement": [
        {
            "Effect": "Allow",
            "Resource": "arn:aws:s3express:region-code:account-id:bucket/DOC-EXAMPLE-
BUCKET",
            "Action": [
                "s3express:CreateSession"
```

```
            ]
        }
    ]

}
```

Buat dan konfigurasikan cluster Anda

Selanjutnya, buat cluster yang menjalankan Spark dengan S3 Express One Zone. Langkah-langkah berikut menjelaskan ikhtisar tingkat tinggi untuk membuat cluster diAWS Management Console:

1. Arahkan ke konsol EMR Amazon dan pilih Clusters dari sidebar. Kemudian pilih Buat cluster.
2. Pilih rilis EMR Amazon `emr-6.15.0` atau yang lebih tinggi.
3. Pilih bundel aplikasi interaktif Spark, dan pilih aplikasi lain yang mungkin ingin Anda sertakan di cluster Anda. Anda harus menyertakan setidaknya Spark dan Hadoop di cluster Anda.
4. Untuk mengaktifkan Amazon S3 Express One Zone, masukkan konfigurasi yang mirip dengan contoh berikut di bagian Pengaturan perangkat lunak. Konfigurasi dan nilai yang direkomendasikan dijelaskan di Ikhtisar konfigurasi bagian yang mengikuti prosedur ini.

   ```
   [
     {
       "Classification": "core-site",
       "Properties": {
         "fs.s3a.aws.credentials.provider":
    "software.amazon.awssdk.auth.credentials.InstanceProfileCredentialsProvider",
         "fs.s3a.change.detection.mode": "none",
         "fs.s3a.endpoint.region": "aa-example-1",
         "fs.s3a.select.enabled": "false"
       }
     },
     {
       "Classification": "spark-defaults",
       "Properties": {
         "spark.sql.sources.fastS3PartitionDiscovery.enabled": "false"
       }
     }
   ]
   ```

5. Di bagian profil instans EC2 untuk Amazon EMR, pilih untuk menggunakan peran yang ada, dan gunakan peran dengan kebijakan terlampir yang Anda buat di bagian di Buat kebijakan izin atas.
6. Konfigurasikan setelan klaster lainnya yang sesuai untuk aplikasi Anda, lalu pilih Buat klaster.

Ikhtisar konfigurasi

Tabel berikut menjelaskan konfigurasi dan nilai yang disarankan yang harus Anda tentukan saat menyiapkan klaster yang menggunakan S3 Express One Zone dengan Amazon EMR, seperti yang dijelaskan di bagian. Buat dan konfigurasikan cluster Anda

S3Akonfigurasi

Parameter	Nilai default	Nilai yang disarankan	Penjelasan
`fs.s3a.aws.credentials.provider`	Jika tidak ditentukan, gunakan `AWSCredentialProviderList` dalam urutan sebagai berikut:`TemporaryAWSCredentialsProvider` ,`SimpleAWSCredentialsProvider` ,`EnvironmentVariableCredentialsProvider` ,`IAMInstanceCredentialsProvider` .	`software.amazon.awssdk.auth.credentials.InstanceProfileCredentialsProvider`	Peran profil instans EMR Amazon harus memiliki kebijakan yang memungkinkan S3A sistem file untuk memanggil . `s3express:CreateSession` Penyedia crendential lainnya juga berfungsi jika mereka memiliki izin S3 Express One Zone.
`fs.s3a.endpoint.region`	kosong	Di Wilayah AWS mana Anda membuat ember.	Logika resolusi wilayah tidak berfungsi dengan kelas penyimpanan S3 Express One Zone.

Parameter	Nilai default	Nilai yang disarankan	Penjelasan
`fs.s3a.select.enabled`	`true`	`false`	Amazon S3 tidak `select` didukung dengan kelas penyimpanan S3 Express One Zone.
`fs.s3a.change.detection.mode`	`server`	tidak ada	Ubah deteksi dengan S3A bekerja dengan memeriksa MD5 berbasis`etags`. Kelas penyimpanan S3 Express One Zone tidak mendukung MD5`checksums` .

Sparkkonfigurasi

Parameter	Nilai default	Nilai yang disarankan	Penjelasan
`spark.sql.sources.fastS3PartitionDiscovery.enabled`	`true`	`false`	Pengoptimalan internal menggunakan parameter API S3 yang tidak didukung oleh kelas penyimpanan S3 Express One Zone.

Pertimbangan-pertimbangan

Pertimbangkan hal berikut saat Anda mengintegrasikan Apache Spark di Amazon EMR dengan kelas penyimpanan S3 Express One Zone:

- Amazon S3 Express One Zone didukung dengan Amazon EMR rilis 6.15.0 dan lebih tinggi.

- Konektor S3A diperlukan untuk menggunakan S3 Express One Zone dengan Amazon EMR. Hanya S3A yang memiliki fitur dan kelas penyimpanan yang diperlukan untuk berinteraksi dengan S3 Express One Zone. Untuk langkah-langkah untuk mengatur konektor, lihatthe section called "Prasyarat".
- Kelas penyimpanan Amazon S3 Express One Zone hanya didukung dengan Spark pada cluster EMR Amazon yang berjalan di Amazon EC2.
- Kelas penyimpanan Amazon S3 Express One Zone hanya mendukung SSE-S3 enkripsi. Untuk informasi selengkapnya, lihat Enkripsi sisi server dengan kunci terkelola Amazon S3 (SSE-S3).
- Kelas penyimpanan Amazon S3 Express One Zone tidak mendukung penulisan dengan S3A. `FileOutputCommitter` Menulis dengan S3A `FileOutputCommitter` pada bucket S3 Express One Zone menghasilkan kesalahan:. InvalidStorageClass: The storage class you specified is not valid
- Kelas penyimpanan Amazon S3 Express One Zone tidak didukung dengan Amazon EMR Tanpa Server atau Amazon EMR di EKS.

Unggah data dengan AWS DataSync

AWS DataSyncadalah layanan transfer data online yang menyederhanakan, mengotomatiskan, dan mempercepat proses pemindahan data antara layanan penyimpanan dan penyimpanan lokal Anda atau di antara layanan AWS penyimpanan. AWS DataSync mendukung berbagai sistem penyimpanan lokal seperti Hadoop Distributed File System (HDFS), server file NAS, dan penyimpanan objek yang dikelola sendiri.

Cara paling umum untuk mendapatkan data ke cluster adalah dengan mengunggah data ke Amazon S3 dan menggunakan fitur bawaan Amazon EMR untuk memuat data ke cluster Anda.

DataSync dapat membantu Anda menyelesaikan tugas-tugas berikut:

- Replikasi HDFS di cluster Hadoop Anda ke Amazon S3 untuk kelangsungan bisnis
- Salin HDFS ke Amazon S3 untuk mengisi data lake Anda
- Transfer data antara HDFS cluster Hadoop Anda dan Amazon S3 untuk analisis dan pemrosesan

Untuk mengunggah data ke bucket S3, Anda terlebih dahulu menerapkan satu atau beberapa DataSync agen di jaringan yang sama dengan penyimpanan lokal. Agen adalah mesin virtual (VM) yang digunakan untuk membaca data dari atau menulis data ke lokasi yang dikelola sendiri. Anda kemudian mengaktifkan agen Anda di Akun AWS dan Wilayah AWS di mana ember S3 Anda berada.

Setelah agen diaktifkan, Anda membuat lokasi sumber untuk penyimpanan lokal, lokasi tujuan untuk bucket S3, dan tugas. Tugas adalah satu set dari dua lokasi (sumber dan tujuan) dan satu set dari opsi default yang Anda gunakan untuk mengontrol perilaku tugas.

Akhirnya, Anda menjalankan DataSync tugas Anda untuk mentransfer data dari sumber ke tujuan.

Untuk informasi selengkapnya, silakan lihat Memulai dengan AWS DataSync.

Impor file dengan cache terdistribusi

Topik

- Tipe file yang didukung
- Lokasi file yang di-cache
- Mengakses file cache dari aplikasi streaming
- Mengakses file cache dari aplikasi streaming

DistributedCacheadalah fitur Hadoop yang dapat meningkatkan efisiensi ketika peta atau tugas pengurangan membutuhkan akses ke data umum. Jika klaster Anda bergantung pada aplikasi atau binari yang ada yang tidak diinstal saat cluster dibuat, Anda dapat menggunakan DistributedCache untuk mengimpor file-file ini. Fitur ini memungkinkan simpul klaster membaca file yang diimpor dari sistem file lokalnya, alih-alih mengambil file dari simpul klaster lainnya.

Untuk informasi lebih lanjut, kunjungi http://hadoop.apache.org/docs/stable/api/org/apache/hadoop/filecache/ DistributedCache .html.

Anda memanggil DistributedCache saat Anda membuat cluster. File di-cache tepat sebelum memulai pekerjaan Hadoop dan file tetap di-cache selama pekerjaan berlangsung. Anda dapat menyimpan file cache pada sistem file yang kompatibel dengan Hadoop, misalnya HDFS atau Amazon S3. Ukuran default cache file adalah 10GB. Untuk mengubah ukuran cache, konfigurasi ulang parameter Hadoop, `local.cache.size` menggunakan tindakan bootstrap. Untuk informasi selengkapnya, lihat Buat tindakan bootstrap untuk menginstal perangkat lunak tambahan.

Tipe file yang didukung

DistributedCachememungkinkan file tunggal dan arsip. File individual di-cache sebagai hanya baca. File yang dapat dieksekusi dan file biner memiliki izin eksekusi yang ditetapkan.

Arsip adalah satu atau lebih file yang dikemas menggunakan utilitas, sepertigzip. DistributedCachemeneruskan file terkompresi ke setiap node inti dan mendekompresi arsip sebagai bagian dari caching. DistributedCachemendukung format kompresi berikut:

- zip
- tgz
- tar.gz
- tar
- jar

Lokasi file yang di-cache

DistributedCachemenyalin file ke node inti saja. Jika tidak ada node inti di cluster, DistributedCache salin file ke node utama.

DistributedCachemengaitkan file cache ke direktori kerja mapper dan peredam saat ini menggunakan symlink. Tautan simbol adalah suatu alias ke lokasi file, bukan lokasi file sebenarnya. Nilai parameter, `yarn.nodemanager.local-dirs` dalam `yarn-site.xml`, menentukan lokasi file sementara. Amazon EMR menetapkan parameter ini ke `/mnt/mapred`, atau beberapa variasi berdasarkan tipe instans dan versi EMR. Misalnya, setelan mungkin memiliki `/mnt/mapred` dan `/mnt1/mapred` karena tipe instans memiliki dua volume sementara. File cache terletak di subdirektori lokasi file sementara di `/mnt/mapred/taskTracker/archive`.

Jika Anda men-cache satu file, DistributedCache letakkan file di `archive` direktori. Jika Anda menyimpan arsip, DistributedCache mendekompresi file, membuat subdirektori dengan nama yang sama `/archive` dengan nama file arsip. File individu terletak di subdirektori baru.

Anda DistributedCache hanya dapat menggunakan saat menggunakan Streaming.

Mengakses file cache dari aplikasi streaming

Untuk mengakses file yang di-cache dari aplikasi pemeta atau peredam Anda, pastikan bahwa Anda telah menambahkan direktori kerja saat ini (./) ke dalam jalur aplikasi Anda dan mereferensikan file yang di-cache seolah-olah ada di direktori kerja saat ini.

Mengakses file cache dari aplikasi streaming

Anda dapat menggunakan AWS Management Console dan AWS CLI untuk membuat cluster yang menggunakan Distributed Cache.

> **Note**
>
> Kami telah mendesain ulang konsol EMR Amazon agar lebih mudah digunakan. Lihat Apa yang baru dengan konsol? untuk mempelajari tentang perbedaan antara pengalaman konsol lama dan baru.

New console

Untuk menentukan file cache terdistribusi dengan konsol baru

1. Masuk keAWS Management Console, dan buka konsol EMR Amazon di https://console.aws.amazon.com/emr.
2. Di bawah EMR pada EC2 di panel navigasi kiri, pilih Clusters, lalu pilih Create cluster.
3. Di bawah Langkah, pilih Tambahkan langkah. Ini membuka dialog Tambah langkah. Di bidang Argumen, sertakan file dan arsip untuk disimpan ke cache. Ukuran file (atau ukuran total file dalam file arsip) harus kurang dari ukuran cache yang dialokasikan.

   Jika Anda ingin menambahkan file individual ke cache terdistribusi`-cacheFile`, tentukan, diikuti dengan nama dan lokasi file, tanda pound (#), dan nama yang ingin Anda berikan file saat ditempatkan di cache lokal. Contoh berikut menunjukkan bagaimana menambahkan file individual ke cache didistribusikan.

   ```
   -cacheFile \
   s3://DOC-EXAMPLE-BUCKET/file-name#cache-file-name
   ```

   Jika Anda ingin menambahkan file arsip ke cache terdistribusi, masukkan `-cacheArchive` diikuti dengan lokasi file di Amazon S3, tanda pound (#), dan kemudian nama yang ingin Anda berikan koleksi file di cache lokal. Contoh berikut menunjukkan bagaimana menambahkan file arsip ke cache didistribusikan.

   ```
   -cacheArchive \
   s3://DOC-EXAMPLE-BUCKET/archive-name#cache-archive-name
   ```

   Masukkan nilai yang sesuai di bidang dialog lainnya. Opsi akan berbeda tergantung pada tipe langkah. Untuk menambahkan langkah Anda dan keluar dari dialog, pilih Tambah langkah.
4. Pilih opsi lain yang berlaku untuk cluster Anda.

5. Untuk meluncurkan klaster Anda, pilih Buat klaster.

Old console

Untuk menentukan file cache terdistribusi dengan konsol lama

1. Arahkan ke konsol EMR Amazon baru dan pilih Beralih ke konsol lama dari navigasi samping. Untuk informasi selengkapnya tentang apa yang diharapkan saat beralih ke konsol lama, lihat Menggunakan konsol lama.
2. Pilih Buat klaster.
3. Pilih Eksekusi langkah sebagai modus Peluncuran.
4. Di bagian Langkah, di bidang Tambahkan langkah, pilih Program streaming dari daftar dan klik Konfigurasi dan tambah.
5. Di bidang Argumen, sertakan file dan arsip untuk disimpan ke cache dan pilih Tambah. Ukuran file (atau ukuran total file dalam file arsip) harus kurang dari ukuran cache yang dialokasikan.

   Jika Anda ingin menambahkan file individual ke cache terdistribusi`-cacheFile`, tentukan, diikuti dengan nama dan lokasi file, tanda pound (#), dan nama yang ingin Anda berikan file saat ditempatkan di cache lokal. Contoh berikut menunjukkan bagaimana menambahkan file individual ke cache didistribusikan.

   ```
   -cacheFile \
   s3://DOC-EXAMPLE-BUCKET/file_name#cache_file_name
   ```

   Jika Anda ingin menambahkan file arsip ke cache terdistribusi, masukkan `-cacheArchive` diikuti dengan lokasi file di Amazon S3, tanda pound (#), dan kemudian nama yang ingin Anda berikan koleksi file di cache lokal. Contoh berikut menunjukkan bagaimana menambahkan file arsip ke cache didistribusikan.

   ```
   -cacheArchive \
   s3://DOC-EXAMPLE-BUCKET/archive_name#cache_archive_name
   ```
6. Lanjutkan dengan mengonfigurasi dan meluncurkan cluster Anda. Klaster Anda menyalin file ke lokasi cache sebelum memproses langkah klaster apa pun.

CLI

Untuk menentukan file cache terdistribusi dengan AWS CLI

- Untuk mengirimkan langkah Streaming saat cluster dibuat, ketik perintah `create-cluster` dengan parameter `--steps`. Untuk menentukan file cache terdistribusi menggunakan AWS CLI, tentukan argumen yang sesuai saat mengirimkan langkah Streaming.

  Jika Anda ingin menambahkan file individual ke cache terdistribusi`-cacheFile`, tentukan, diikuti dengan nama dan lokasi file, tanda pound (#), dan nama yang ingin Anda berikan file saat ditempatkan di cache lokal.

  Jika Anda ingin menambahkan file arsip ke cache terdistribusi, masukkan `-cacheArchive` diikuti dengan lokasi file di Amazon S3, tanda pound (#), dan kemudian nama yang ingin Anda berikan koleksi file di cache lokal. Contoh berikut menunjukkan bagaimana menambahkan file arsip ke cache didistribusikan.

  Untuk informasi selengkapnya tentang menggunakan perintah Amazon EMR di AWS CLI, lihat https://docs.aws.amazon.com/cli/latest/reference/emr.

Example 1

Ketik perintah berikut untuk meluncurkan klaster dan mengirimkan langkah Streaming yang digunakan `-cacheFile` untuk menambahkan satu file, `sample_dataset_cached.dat`, ke cache.

```
aws emr create-cluster --name "Test cluster" --release-label emr-4.0.0 --
applications Name=Hive Name=Pig --use-default-roles --ec2-attributes KeyName=myKey
 --instance-type m5.xlarge --instance-count 3 --steps Type=STREAMING,Name="Streaming
 program",ActionOnFailure=CONTINUE,Args=["--files","s3://my_bucket/my_mapper.py
 s3://my_bucket/my_reducer.py","-mapper","my_mapper.py","-reducer","my_reducer.py,"-
input","s3://my_bucket/my_input","-output","s3://my_bucket/my_output", "-
cacheFile","s3://my_bucket/sample_dataset.dat#sample_dataset_cached.dat"]
```

Saat Anda menentukan jumlah instance tanpa menggunakan `--instance-groups` parameter, satu node primer diluncurkan, dan instance yang tersisa diluncurkan sebagai node inti. Semua simpul akan menggunakan tipe instans yang ditentukan dalam perintah.

Jika Anda belum pernah membuat peran layanan EMR default dan profil instans EC2, ketik `aws emr create-default-roles` untuk membuatnya sebelum mengetik subperintah `create-cluster`.

Example 2

Perintah berikut menunjukkan pembuatan klaster streaming dan menggunakan `-cacheArchive` untuk menambahkan arsip file ke cache.

```
aws emr create-cluster --name "Test cluster" --release-label emr-4.0.0 --
applications Name=Hive Name=Pig --use-default-roles --ec2-attributes KeyName=myKey
 --instance-type m5.xlarge --instance-count 3 --steps Type=STREAMING,Name="Streaming
 program",ActionOnFailure=CONTINUE,Args=["--files","s3://my_bucket/my_mapper.py
 s3://my_bucket/my_reducer.py","-mapper","my_mapper.py","-reducer","my_reducer.py","-
input","s3://my_bucket/my_input","-output","s3://my_bucket/my_output", "-
cacheArchive","s3://my_bucket/sample_dataset.tgz#sample_dataset_cached"]
```

Saat Anda menentukan jumlah instance tanpa menggunakan `--instance-groups` parameter, satu node primer diluncurkan, dan instance yang tersisa diluncurkan sebagai node inti. Semua simpul akan menggunakan tipe instans yang ditentukan dalam perintah.

Jika Anda belum pernah membuat peran layanan EMR default dan profil instans EC2, ketik `aws emr create-default-roles` untuk membuatnya sebelum mengetik subperintah `create-cluster`.

Cara memproses file terkompresi

Hadoop memeriksa ekstensi file untuk mendeteksi file terkompresi. Jenis kompresi yang didukung oleh Hadoop adalah: gzip, bzip2, dan LZO. Anda tidak perlu melakukan tindakan tambahan apa pun untuk mengekstrak file jika jenis kompresi ini digunakan; Hadoop menanganinya untuk Anda.

Untuk mengindeks file LZO, Anda dapat menggunakan pustaka hadoop-lzo yang dapat diunduh dari https://github.com/kevinweil/hadoop-lzo. Perhatikan bahwa karena ini merupakan pustaka pihak ketiga, Amazon EMR tidak menawarkan dukungan developer tentang cara menggunakan alat ini. Untuk informasi penggunaan, lihat the hadoop-lzo readme file.

Mengimpor data DynamoDB ke Hive

Implementasi Hive yang disediakan oleh Amazon EMR mencakup fungsionalitas yang dapat Anda gunakan untuk mengimpor dan mengekspor data antara DynamoDB dan klaster Amazon EMR. Ini

berguna jika data input Anda disimpan di DynamoDB. Untuk informasi selengkapnya, lihat Ekspor, Impor, kueri, dan menggabungkan tabel di DynamoDB menggunakan Amazon EMR.

Connect ke data dengan AWS Direct Connect

AWS Direct Connect adalah layanan yang dapat Anda gunakan untuk membuat sambungan jaringan khusus pribadi ke Amazon Web Services dari pusat data, kantor, atau lingkungan kolokasi Anda. Jika Anda memiliki sejumlah besar input data, dengan menggunakan AWS Direct Connect, Anda dapat mengurangi biaya jaringan, meningkatkan bandwidth throughput, dan memberikan pengalaman jaringan yang lebih konsisten daripada koneksi berbasis Internet. Untuk informasi selengkapnya, lihat AWS Direct Connect Panduan Pengguna.

Unggah data dalam jumlah besar dengan AWS Snowball

AWS Snowballadalah layanan yang dapat Anda gunakan untuk mentransfer data dalam jumlah besar antara Amazon Simple Storage Service (Amazon S3) dan lokasi penyimpanan data di tempat Anda dengan kecepatan tinggi. faster-than-internet Snowball mendukung dua jenis pekerjaan: pekerjaan impor dan pekerjaan ekspor. Pekerjaan impor melibatkan transfer data dari sumber on-premise ke bucket Amazon S3. Pekerjaan ekspor melibatkan transfer data dari bucket Amazon S3 ke sumber on-Premise. Untuk kedua jenis pekerjaan, perangkat Snowball mengamankan dan melindungi data Anda sementara operator pengiriman regional mengangkutnya antara Amazon S3 dan lokasi penyimpanan data di tempat Anda. Perangkat Snowball secara fisik kokoh dan dilindungi oleh AWS Key Management Service (AWS KMS). Untuk informasi selengkapnya, lihat AWS Snowball Panduan Developer Edge.

Mengkonfigurasi lokasi output

Format output paling umum dari klaster Amazon EMR adalah sebagai file teks, baik yang dikompresi atau tidak dikompresi. Biasanya, ini ditulis ke bucket Amazon S3. Bucket ini harus dibuat sebelum Anda meluncurkan klaster. Anda menentukan S3 bucket sebagai lokasi output ketika Anda memulai klaster.

Untuk informasi selengkapnya, lihat topik berikut:

Topik

- Buat dan konfigurasikan bucket Amazon S3
- Format apa yang dapat dikembalikan oleh Amazon EMR?
- Cara menulis data ke bucket Amazon S3 yang tidak Anda miliki
- Kompres output klaster Anda

Buat dan konfigurasikan bucket Amazon S3

Amazon EMR (Amazon EMR) menggunakan Amazon S3 untuk menyimpan data input, berkas log, dan data output. Amazon S3 mengacu pada lokasi penyimpanan ini sebagai bucket. Bucket memiliki pembatasan dan batasan tertentu agar sesuai dengan persyaratan Amazon S3 dan DNS. Untuk informasi lebih lanjut, kunjungi Pembatasan dan Batasan Bucket dalam Panduan Developer Amazon Simple Storage Service.

Untuk membuat bucket Amazon S3, ikuti petunjuk di halaman Membuat bucket dalam Panduan Developer Amazon Simple Storage Service.

> **Note**
>
> Jika Anda mengaktifkan pencatatan log di panduan Membuat Bucket, ini hanya mengaktifkan log akses bucket, bukan log cluster.

> **Note**
>
> Untuk informasi lebih lanjut tentang menentukan bucket khusus Wilayah, lihat Bucket dan Wilayah di Panduan Developer Amazon Simple Storage Service dan Titik Akhir Wilayah yang Tersedia untuk AWS SDK .

Setelah Anda membuat bucket, Anda dapat mengatur izin yang sesuai terhadapnya. Biasanya, Anda memberi diri Anda (pemilik) akses baca dan tulis. Kami sangat menyarankan agar Anda mengikuti Praktik Terbaik Keamanan untuk Amazon S3 saat mengonfigurasi bucket Anda.

Bucket Amazon S3 yang diperlukan harus ada sebelum Anda dapat membuat klaster. Anda harus mengunggah skrip atau data yang diperlukan yang dimaksud dalam klaster ke Amazon S3. Tabel berikut menjelaskan contoh data, skrip, dan lokasi berkas log.

Informasi	Contoh Lokasi di Amazon S3
skrip atau program	`s3://DOC-EXAMPLE-BUCKET1 /script/MapperScript.py`
berkas log	`s3://DOC-EXAMPLE-BUCKET1 /logs`

Informasi	Contoh Lokasi di Amazon S3
data input	`s3://DOC-EXAMPLE-BUCKET1/input`
data output	`s3://DOC-EXAMPLE-BUCKET1/output`

Format apa yang dapat dikembalikan oleh Amazon EMR?

Format output default untuk klaster adalah teks dengan kunci, pasangan nilai yang ditulis ke baris individual dari file teks. Ini adalah format output yang paling umum digunakan.

Jika data output Anda harus ditulis dalam format selain file teks default, Anda dapat menggunakan `OutputFormat` antarmuka Hadoop untuk menentukan jenis output lainnya. Anda bahkan dapat membuat subkelas dari kelas `FileOutputFormat` untuk menangani tipe data khusus. Untuk informasi lebih lanjut, lihat http://hadoop.apache.org/docs/current/api/org/apache/hadoop/mapred/OutputFormat .html.

Jika Anda meluncurkan cluster Hive, Anda dapat menggunakan serializer/deserializer (SerDe) untuk mengeluarkan data dari HDFS ke format tertentu. Untuk informasi lebih lanjut, lihat https://cwiki.apache.org/confluence/display/Hive/ SerDe.

Cara menulis data ke bucket Amazon S3 yang tidak Anda miliki

Saat Anda menulis file ke bucket Amazon Simple Storage Service (Amazon S3), secara default, hanya Anda yang dapat membaca file tersebut. Asumsinya adalah bahwa Anda akan menulis file ke bucket Anda sendiri, dan pengaturan default ini melindungi privasi file Anda.

Namun, jika Anda menjalankan sebuah klaster, dan Anda ingin output ditulis ke bucket Amazon S3 pengguna AWS lain, dan Anda ingin pengguna AWS lain itu dapat membaca output tersebut, Anda harus melakukan dua hal:

- Minta pengguna AWS lain memberi Anda izin menulis untuk bucket Amazon S3 mereka. Klaster yang Anda luncurkan berjalan di bawah kredensial AWS Anda, sehingga setiap klaster yang Anda luncurkan juga dapat menulis ke bucket pengguna AWS lain tersebut.
- Tetapkan izin baca untuk pengguna AWS lain pada file yang Anda atau klaster tulis ke bucket Amazon S3. Cara termudah untuk menetapkan izin baca ini adalah dengan menggunakan daftar kontrol akses (ACL) terekam, satu set kebijakan akses yang telah ditentukan sebelumnya yang ditentukan oleh Amazon S3.

Untuk informasi tentang cara AWS pengguna lain dapat memberi Anda izin untuk menulis file ke bucket Amazon S3 pengguna lain, lihat Mengedit izin bucket di Panduan Pengguna Layanan Penyimpanan Sederhana Amazon.

Agar klaster Anda menggunakan ACL terekam saat menulis file ke Amazon S3, atur opsi konfigurasi klaster `fs.s3.canned.acl` ke ACL terekam yang akan digunakan. Tabel berikut mencantumkan ACL terekam yang ditentukan saat ini.

ACL Terekam	Deskripsi
`AuthenticatedRead`	Menentukan bahwa pemilik diberikan `Permission.FullControl` dan penerima grup `GroupGrantee.AuthenticatedUsers` diberikan akses `Permission.Read`.
`BucketOwnerFullControl`	Menentukan bahwa pemilik bucket diberikan `Permission.FullControl`. Pemilik bucket belum tentu sama dengan pemilik objek.
`BucketOwnerRead`	Menentukan bahwa pemilik bucket diberikan `Permission.Read`. Pemilik bucket belum tentu sama dengan pemilik objek.
`LogDeliveryWrite`	Menentukan bahwa pemilik diberikan `Permission.FullControl` dan penerima grup `GroupGrantee.LogDelivery` diberikan akses `Permission.Write`, sehingga log akses dapat dikirim.
`Private`	Menentukan bahwa pemilik diberikan `Permission.FullControl`.
`PublicRead`	Menentukan bahwa pemilik diberikan`Permission.FullControl` dan penerima grup `GroupGrantee.AllUsers` diberikan akses `Permission.Read`.
`PublicReadWrite`	Menentukan bahwa pemilik diberikan `Permission.FullControl` dan penerima grup `GroupGran`

ACL Terekam	Deskripsi
	`tee.AllUsers` diberikan `Permission.Read` dan akses `Permission.Write` .

Terdapat berbagai cara untuk mengatur opsi konfigurasi klaster, tergantung pada jenis klaster yang Anda jalankan. Prosedur berikut menunjukkan cara mengatur opsi untuk kasus umum.

Untuk menulis file menggunakan ACL terekam di Hive

- Dari prompt perintah Hive, atur opsi konfigurasi `fs.s3.canned.acl` ke ACL terekam yang Anda inginkan agar klaster diatur pada file yang ditulisnya ke Amazon S3. Untuk mengakses prompt perintah Hive, sambungkan ke simpul utama menggunakan SSH, dan ketik Hive di prompt perintah Hadoop. Untuk informasi selengkapnya, lihat Connect ke node utama menggunakan SSH.

  Contoh berikut mengatur opsi konfigurasi `fs.s3.canned.acl` ke `BucketOwnerFullControl`, yang memberi pemilik bucket Amazon S3 kendali penuh atas file tersebut. Perhatikan bahwa perintah set peka terhadap huruf besar-kecil dan tidak mengandung tanda kutip atau spasi.

  ```
  hive> set fs.s3.canned.acl=BucketOwnerFullControl;
  create table acl (n int) location 's3://acltestbucket/acl/';
  insert overwrite table acl select count(*) from acl;
  ```

  Dua baris terakhir dari contoh membuat tabel yang disimpan di Amazon S3 dan menulis data ke tabel.

Untuk menulis file menggunakan ACL terekam di Pig

- Dari prompt perintah Pig, atur opsi konfigurasi `fs.s3.canned.acl` ke ACL terekam yang Anda inginkan agar klaster diatur pada file yang ditulisnya ke Amazon S3. Untuk mengakses prompt perintah Pig, sambungkan ke simpul utama menggunakan SSH, dan ketik Pig pada prompt perintah Hadoop. Untuk informasi selengkapnya, lihat Connect ke node utama menggunakan SSH.

Contoh berikut menyetel opsi `fs.s3.canned.acl` konfigurasi BucketOwnerFullControl, yang memberi pemilik bucket Amazon S3 kontrol penuh atas file tersebut. Perhatikan bahwa perintah set menyertakan satu spasi sebelum nama ACL terekam dan tidak berisi tanda kutip.

```
pig> set fs.s3.canned.acl BucketOwnerFullControl;
store some data into 's3://acltestbucket/pig/acl';
```

Untuk menulis file menggunakan ACL terekam dalam JAR kustom

- Atur opsi konfigurasi `fs.s3.canned.acl` menggunakan Hadoop dengan menggunakan bendera -D. Ini ditunjukkan dalam contoh di bawah.

```
hadoop jar hadoop-examples.jar wordcount
-Dfs.s3.canned.acl=BucketOwnerFullControl s3://mybucket/input s3://mybucket/output
```

Kompres output klaster Anda

Topik

- Kompresi data output
- Kompresi data menengah
- Menggunakan pustaka Snappy dengan Amazon EMR

Kompresi data output

Ini mengompres output dari pekerjaan Hadoop Anda. Jika Anda menggunakan TextOutputFormat hasilnya adalah file teks gzip'ed. Jika Anda menulis untuk SequenceFiles maka hasilnya adalah SequenceFile yang dikompresi secara internal. Hal ini dapat diaktifkan dengan menetapkan pengaturan konfigurasi mapred.output.compress ke betul.

Jika Anda menjalankan tugas streaming, Anda dapat mengaktifkan ini dengan memasukkan tugas streaming argumen ini.

```
-jobconf mapred.output.compress=true
```

Anda juga dapat menggunakan tindakan bootstrap untuk secara otomatis mengompresi semua output tugas. Berikut adalah cara melakukannya dengan klien Ruby.

```
--bootstrap-actions s3://elasticmapreduce/bootstrap-actions/configure-hadoop \
--args "-s,mapred.output.compress=true"
```

Terakhir, jika sedang menulis Jar Kustom, Anda dapat mengaktifkan kompresi output dengan baris berikut saat membuat tugas Anda.

```
FileOutputFormat.setCompressOutput(conf, true);
```

Kompresi data menengah

Jika tugas Anda mengacak sejumlah besar data dari pemeta hingga peredam, Anda dapat melihat peningkatan kinerja dengan mengaktifkan kompresi menengah. Kompres output peta dan dekompresi ketika tiba di simpul inti. Pengaturan konfigurasinya adalah mapred.compress.map.output. Anda dapat mengaktifkan ini yang mana serupa dengan kompresi output.

Saat menulis Jar Kustom, gunakan perintah berikut:

```
conf.setCompressMapOutput(true);
```

Menggunakan pustaka Snappy dengan Amazon EMR

Snappy adalah pustaka kompresi dan dekompresi yang dioptimalkan untuk kecepatan. Pustaka ini tersedia di Amazon EMR AMIS versi 2.0 dan yang lebih baru dan digunakan sebagai default untuk kompresi menengah. Untuk informasi selengkapnya tentang Snappy, kunjungi http://code.google.com/p/snappy/.

Rencanakan dan konfigurasikan node primer

Saat meluncurkan klaster EMR Amazon, Anda dapat memilih untuk memiliki satu atau tiga node utama di cluster Anda. Ketersediaan tinggi untuk armada misalnya didukung dengan rilis Amazon EMR 5.36.1, 6.8.1, 6.9.1, 6.10.1, 6.11.1, 6.12.0, dan yang lebih tinggi. Misalnya grup, ketersediaan tinggi didukung dengan rilis Amazon EMR 5.23.0 dan yang lebih tinggi. Untuk lebih meningkatkan ketersediaan klaster, Amazon EMR dapat menggunakan grup penempatan Amazon EC2 untuk memastikan bahwa node utama ditempatkan pada perangkat keras dasar yang berbeda. Untuk informasi selengkapnya, lihat Integrasi Amazon EMR dengan grup penempatan EC2.

Cluster EMR Amazon dengan beberapa node primer memberikan manfaat berikut:

- Node primer tidak lagi menjadi titik kegagalan tunggal. Jika salah satu node primer gagal, cluster menggunakan dua node primer lainnya dan berjalan tanpa gangguan. Sementara itu, Amazon EMR secara otomatis menggantikan node primer yang gagal dengan yang baru yang disediakan dengan konfigurasi dan tindakan bootstrap yang sama.
- Amazon EMR memungkinkan Hadoop fitur ketersediaan tinggi HDFS NameNode dan YARN ResourceManager dan mendukung ketersediaan tinggi untuk beberapa aplikasi open source lainnya.

  Untuk informasi selengkapnya tentang bagaimana klaster EMR Amazon dengan beberapa node utama mendukung aplikasi open source dan fitur EMR Amazon lainnya, lihat. Aplikasi dan fitur yang didukung

> **Note**
>
> Klaster hanya dapat berada di satu Availability Zone atau subnet.

Bagian ini memberikan informasi tentang aplikasi dan fitur yang didukung dari klaster EMR Amazon dengan beberapa node utama serta detail konfigurasi, praktik terbaik, dan pertimbangan untuk meluncurkan cluster.

Topik

- Aplikasi dan fitur yang didukung
- Luncurkan Amazon EMR Cluster dengan beberapa node utama
- Integrasi Amazon EMR dengan grup penempatan EC2

- Pertimbangan dan praktik terbaik

Aplikasi dan fitur yang didukung

Topik ini memberikan informasi tentang fitur ketersediaan tinggi Hadoop HDFS NameNode dan YARN di cluster EMR ResourceManager Amazon, dan bagaimana fitur ketersediaan tinggi bekerja dengan aplikasi open source dan fitur EMR Amazon lainnya.

HDFS ketersediaan tinggi

Cluster EMR Amazon dengan beberapa node utama memungkinkan fitur ketersediaan HDFS NameNode tinggi di Hadoop. Untuk informasi lebih lanjut, lihat ketersediaan tinggi HDFS.

Dalam cluster EMR Amazon, dua atau lebih node terpisah dikonfigurasi sebagai. NameNodes Yang satu NameNode berada di `active` negara bagian dan yang lainnya dalam `standby` keadaan. Jika node `active` NameNode gagal, Amazon EMR memulai proses failover HDFS otomatis. Sebuah node dengan `standby` NameNode menjadi `active` dan mengambil alih semua operasi klien di cluster. Amazon EMR menggantikan node yang gagal dengan yang baru, yang kemudian bergabung kembali sebagai file. `standby`

> **Note**
>
> Di Amazon EMR versi 5.23.0 hingga dan termasuk 5.30.1, hanya dua dari tiga node utama yang menjalankan HDFS. NameNode

Jika Anda perlu mencari tahu yang NameNode mana`active`, Anda dapat menggunakan SSH untuk terhubung ke node utama apa pun di cluster dan menjalankan perintah berikut:

```
hdfs haadmin -getAllServiceState
```

Output mencantumkan node tempat NameNode diinstal dan statusnya. Misalnya,

```
ip-##-#-#-##1.ec2.internal:8020 active
ip-##-#-#-##2.ec2.internal:8020 standby
ip-##-#-#-##3.ec2.internal:8020 standby
```

Benang ketersediaan tinggi ResourceManager

Cluster EMR Amazon dengan beberapa node utama memungkinkan fitur ketersediaan ResourceManager tinggi YARN di Hadoop. Untuk informasi selengkapnya, lihat ketersediaan ResourceManager tinggi.

Dalam klaster EMR Amazon dengan beberapa node primer, YARN ResourceManager berjalan pada ketiga node utama. Satu ResourceManager dalam `active` keadaan, dan dua lainnya dalam `standby` keadaan. Jika node utama `active` ResourceManager gagal, Amazon EMR memulai proses failover otomatis. Sebuah node primer dengan `standby` ResourceManager mengambil alih semua operasi. Amazon EMR menggantikan simpul primer yang gagal dengan yang baru, yang kemudian bergabung kembali dengan kuorum sebagai. ResourceManager `standby`

Anda dapat terhubung ke “http: //:8088/cluster *`master-public-dns-name`*” untuk node utama apa pun, yang secara otomatis mengarahkan Anda ke manajer sumber daya. `active` Untuk mengetahui manajer sumber daya mana`active`, gunakan SSH untuk terhubung ke node utama apa pun di cluster. Kemudian jalankan perintah berikut untuk mendapatkan daftar tiga node utama dan statusnya:

```
yarn rmadmin -getAllServiceState
```

Aplikasi yang didukung di Amazon EMR Cluster dengan beberapa node utama

Anda dapat menginstal dan menjalankan aplikasi berikut pada cluster EMR Amazon dengan beberapa node utama. Untuk setiap aplikasi, proses failover node primer bervariasi.

Aplikasi	Ketersediaan selama failover node primer	Catatan
Flink	Ketersediaan tidak terpengaruh oleh failover node primer	Tugas Flink di Amazon EMR dijalankan sebagai aplikasi YARN. Flink JobManagers dijalankan sebagai YARN ApplicationMasters pada node inti. JobManager Ini tidak terpengaruh oleh proses failover node primer. Jika Anda menggunakan Amazon EMR versi 5.27.0 atau lebih lama, ini JobManager adalah satu titik kegagalan. Ketika JobManager

Aplikasi	Ketersediaan selama failover node primer	Catatan
		gagal, ia kehilangan semua status pekerjaan dan tidak akan melanjutkan pekerjaan yang sedang berjalan. Anda dapat mengaktifkan ketersediaan JobManager tinggi dengan mengonfigurasi jumlah upaya aplikasi, pos pemeriksaan, dan mengaktifkan ZooKeeper sebagai penyimpanan status untuk Flink. Untuk informasi selengkapnya, lihat Mengonfigurasi Flink di Cluster EMR Amazon dengan beberapa node utama. Dimulai dengan Amazon EMR versi 5.28.0, tidak diperlukan konfigurasi manual untuk mengaktifkan ketersediaan tinggi. JobManager
Ganglia	Ketersediaan tidak terpengaruh oleh failover node primer	Ganglia tersedia di semua node primer, sehingga Ganglia dapat terus berjalan selama proses failover node primer.
Hadoop	Ketersediaan yang tinggi	HDFS NameNode dan YARN ResourceManager secara otomatis gagal ke node siaga ketika node primer aktif gagal.
HBase	Ketersediaan yang tinggi	HBase secara otomatis gagal ke node siaga ketika node primer aktif gagal. Jika Anda terhubung ke HBase melalui server REST atau Thrift, Anda harus beralih ke node primer yang berbeda ketika node primer aktif gagal.
HCatalog	Ketersediaan tidak terpengaruh oleh failover node primer	HCatalog dibangun di atas metastore Hive, yang ada di luar klaster. HCatalog tetap tersedia selama proses failover node utama.

Aplikasi	Ketersediaan selama failover node primer	Catatan
JupyterHub	Ketersediaan yang tinggi	JupyterHub diinstal pada ketiga instance utama. Sangat disarankan untuk mengonfig urasi persistensi notebook untuk mencegah hilangnya notebook pada kegagalan node primer. Untuk informasi selengkapnya, lihat Mengkonfigurasi persistensi notebook di Amazon S3.
Hidup	Ketersediaan yang tinggi	Livy diinstal pada ketiga node utama. Ketika node primer aktif gagal, Anda kehilangan akses ke sesi Livy saat ini dan perlu membuat sesi Livy baru pada node primer yang berbeda atau pada node pengganti baru.
Mahout	Ketersediaan tidak terpengar uh oleh failover node primer	Karena Mahout tidak memiliki daemon, itu tidak terpengaruh oleh proses failover node utama.
MxNet	Ketersediaan tidak terpengar uh oleh failover node primer	Karena MXNet tidak memiliki daemon, itu tidak terpengaruh oleh proses failover node utama.
Phoenix	Ketersediaan Yang Tinggi	Phoenix' QueryServer berjalan hanya pada salah satu dari tiga node utama. Phoenix pada ketiga master dikonfigurasi untuk menghubun gkan Phoenix QueryServer. Anda dapat menemukan IP pribadi server Phoenix Query dengan menggunakan file `/etc/phoenix/conf/phoenix-env.sh`
Babi	Ketersediaan tidak terpengar uh oleh failover node primer	Karena Babi tidak memiliki daemon, itu tidak terpengaruh oleh proses failover node primer.

Aplikasi	Ketersediaan selama failover node primer	Catatan
Percikan	Ketersediaan yang tinggi	Semua aplikasi Spark berjalan dalam wadah YARN dan dapat bereaksi terhadap failover node primer dengan cara yang sama seperti fitur YARN ketersediaan tinggi.
Sqoop	Ketersediaan yang tinggi	Secara default, sqoop-job dan sqoop-metastore menyimpan data (deskripsi tugas) pada disk lokal utama yang menjalankan perintah, jika Anda ingin menyimpan data metastore di Basis Data eksternal, lihat dokumentasi Apache Sqoop
Tez	Ketersediaan yang tinggi	Karena kontainer Tez berjalan di YARN, Tez berperilaku dengan cara yang sama seperti YARN selama proses failover node utama.
TensorFlow	Ketersediaan tidak terpengaruh oleh failover node primer	Karena tidak TensorFlow memiliki daemon, itu tidak terpengaruh oleh proses failover node utama.
Zeppelin	Ketersediaan yang tinggi	Zeppelin diinstal pada ketiga node utama. Zeppelin menyimpan catatan dan konfigurasi interperter dalam HDFS secara default untuk mencegah kehilangan data. Sesi penerjemah benar-benar terisolasi di ketiga contoh utama. Data sesi akan hilang saat utama mengalami gagal. Disarankan untuk tidak memodifikasi catatan yang sama secara bersamaan pada instance primer yang berbeda.

Aplikasi	Ketersediaan selama failover node primer	Catatan
ZooKeeper	Ketersediaan yang tinggi	ZooKeeper adalah dasar dari fitur failover otomatis HDFS. ZooKeeper menyediakan layanan yang sangat tersedia untuk memelihara data koordinasi, memberi tahu klien tentang perubahan dalam data itu, dan memantau klien untuk kegagalan. Untuk informasi selengkapnya, lihat Failover otomatis HDFS.

Untuk menjalankan aplikasi berikut di kluster EMR Amazon dengan beberapa node utama, Anda harus mengonfigurasi database eksternal. Database eksternal ada di luar cluster dan membuat data persisten selama proses failover node primer. Untuk aplikasi berikut, komponen layanan akan secara otomatis pulih selama proses failover node primer, tetapi pekerjaan aktif mungkin gagal dan perlu dicoba lagi.

Aplikasi	Ketersediaan selama failover node primer	Catatan
Sarang	Ketersediaan tinggi hanya untuk komponen layanan	Metastore eksternal untuk Hive diperlukan. Ini harus berupa metastore eksternal MySQL, karena PostgreSQL tidak didukung untuk cluster multi-master. Untuk informasi selengkapnya, lihat Mengkonfigurasi metastore eksternal untuk Hive.
Rona	Ketersediaan tinggi hanya untuk komponen layanan	Diperlukan basis data eksternal untuk Hue. Untuk informasi selengkapnya, lihat Menggunakan Hue dengan basis data jarak jauh di Amazon RDS.
Oozie	Ketersediaan tinggi hanya untuk komponen layanan	Basis data eksternal untuk Oozie diperlukan. Untuk informasi selengkapnya, lihat

Aplikasi	Ketersediaan selama failover node primer	Catatan
		Menggunakan Oozie dengan basis data jarak jauh di Amazon RDS. Oozie-server dan oozie-client diinstal pada ketiga node utama. Klien oozie dikonfigurasi untuk menyambungkan ke server oozie yang benar secara default.
PrestoDB atau PrestoSQL/ Trino	Ketersediaan tinggi hanya untuk komponen layanan	Metastore Hive eksternal untuk PrestoDB (PrestoSQL di Amazon EMR 6.1.0-6.3.0 atau Trino di Amazon EMR 6.4.0 dan yang lebih baru) diperlukan. Anda dapat menggunakan Presto dengan AWS Katalog Data Glue atau gunakan basis data MySQL eksternal untuk Hive. CLI Presto diinstal pada ketiga node utama sehingga Anda dapat menggunakannya untuk mengakses Koordinator Presto dari salah satu node utama. Koordinator Presto diinstal hanya pada satu node utama. Anda dapat menemukan nama DNS dari node utama tempat Koordinator Presto diinstal dengan memanggil Amazon EMR `describe-cluster` API dan membaca nilai bidang yang dikembalikan dalam respons. `MasterPublicDnsName`

> **Note**
>
> Ketika node utama gagal, Java Database Connectivity (JDBC) atau Open Database Connectivity (ODBC) mengakhiri koneksi ke node utama. Anda dapat terhubung ke salah satu node primer yang tersisa untuk melanjutkan pekerjaan Anda karena daemon metastore

Hive berjalan di semua node utama. Atau Anda bisa menunggu node primer yang gagal diganti.

Bagaimana fitur Amazon EMR bekerja di cluster dengan beberapa node utama

Menghubungkan ke node primer menggunakan SSH

Anda dapat terhubung ke salah satu dari tiga node utama di kluster EMR Amazon menggunakan SSH dengan cara yang sama seperti Anda terhubung ke satu simpul utama. Untuk informasi selengkapnya, lihat Connect to the primary node menggunakan SSH.

Jika node primer gagal, koneksi SSH Anda ke node utama berakhir. Untuk melanjutkan pekerjaan Anda, Anda dapat terhubung ke salah satu dari dua node utama lainnya. Atau, Anda dapat mengakses simpul utama baru setelah Amazon EMR menggantikan yang gagal dengan yang baru.

> **Note**
>
> Alamat IP pribadi untuk node primer pengganti tetap sama dengan yang sebelumnya. Alamat IP publik untuk node primer pengganti dapat berubah. Anda dapat mengambil alamat IP baru di konsol atau dengan menggunakan perintah `describe-cluster` di CLI AWS. NameNode hanya berjalan pada dua node utama. Namun, Anda dapat menjalankan perintah `hdfs` CLI dan mengoperasikan pekerjaan untuk mengakses HDFS pada ketiga node utama.

Bekerja dengan langkah-langkah di Amazon EMR Cluster dengan beberapa node utama

Anda dapat mengirimkan langkah-langkah ke klaster EMR Amazon dengan beberapa node primer dengan cara yang sama seperti Anda bekerja dengan langkah-langkah dalam klaster dengan satu simpul utama. Untuk informasi selengkapnya, lihat Mengirim pekerjaan ke klaster.

Berikut ini adalah pertimbangan untuk bekerja dengan langkah-langkah dalam klaster EMR Amazon dengan beberapa node utama:

- Jika node primer gagal, langkah-langkah yang berjalan pada node primer ditandai sebagai GAGAL. Setiap data yang ditulis secara lokal akan hilang. Namun, status GAGAL mungkin tidak mencerminkan keadaan sebenarnya dari langkah-langkah tersebut.
- Jika langkah yang sedang berjalan telah memulai aplikasi YARN ketika node utama gagal, langkah tersebut dapat berlanjut dan berhasil karena failover otomatis dari node primer.

- Disarankan agar Anda memeriksa status langkah dengan mengacu pada output tugas. Misalnya, MapReduce pekerjaan menggunakan _SUCCESS file untuk menentukan apakah pekerjaan berhasil diselesaikan.
- Disarankan agar Anda menyetel ActionOnFailure parameter ke CONTINUE, atau CANCEL_AND_WAIT, bukan TERMINATE_JOB_FLOW, atau TERMINATE_CLUSTER.

Perlindungan penghentian otomatis

Amazon EMR secara otomatis mengaktifkan perlindungan terminasi untuk semua cluster dengan beberapa node utama, dan mengganti pengaturan eksekusi langkah apa pun yang Anda berikan saat membuat klaster. Anda dapat menonaktifkan perlindungan terminasi setelah cluster diluncurkan. Lihat Mengonfigurasi perlindungan pengakhiran untuk menjalankan klaster. Untuk mematikan klaster dengan beberapa node primer, Anda harus terlebih dahulu memodifikasi atribut cluster untuk menonaktifkan perlindungan terminasi. Untuk instruksi, lihat Mengakhiri Cluster EMR Amazon dengan beberapa node utama.

Untuk informasi selengkapnya tentang perlindungan penghentian, lihat Menggunakan perlindungan pengakhiran.

Fitur yang tidak didukung di Amazon EMR Cluster dengan beberapa node utama

Fitur EMR Amazon berikut saat ini tidak tersedia di kluster EMR Amazon dengan beberapa node utama:

- EMR Notebooks
- Akses sekali klik ke server riwayat Spark yang persisten
- Antarmuka pengguna aplikasi persisten
- Akses sekali klik ke antarmuka pengguna aplikasi persisten saat ini tidak tersedia untuk kluster EMR Amazon dengan beberapa node utama atau untuk Amazon EMRClusters yang terintegrasi dengan Lake Formation. AWS

> **Note**
>
> Untuk menggunakan otentikasi Kerberos di klaster Anda, Anda harus mengonfigurasi KDC eksternal.

Dimulai dengan Amazon EMR versi 5.27.0, Anda dapat mengonfigurasi enkripsi Transparan HDFS pada kluster EMR Amazon dengan beberapa node utama. Untuk informasi selengkapnya, lihat Enkripsi transparan dalam HDFS di Amazon EMR.

Luncurkan Amazon EMR Cluster dengan beberapa node utama

Topik ini memberikan detail konfigurasi dan contoh untuk meluncurkan klaster EMR Amazon dengan beberapa node utama.

> **Note**
>
> Amazon EMR secara otomatis mengaktifkan perlindungan terminasi untuk semua cluster yang memiliki beberapa node utama, dan mengganti setelan penghentian otomatis apa pun yang Anda berikan saat membuat klaster. Untuk mematikan klaster dengan beberapa node primer, Anda harus terlebih dahulu memodifikasi atribut cluster untuk menonaktifkan perlindungan terminasi. Untuk petunjuk, silakan lihat Mengakhiri Cluster EMR Amazon dengan beberapa node utama.

Prasyarat

- Anda dapat meluncurkan kluster EMR Amazon dengan beberapa node utama di subnet VPC publik dan pribadi. EC2-Classic Tidak didukung. Untuk meluncurkan klaster EMR Amazon dengan beberapa node primer di subnet publik, Anda harus mengaktifkan instance di subnet ini untuk menerima alamat IP publik dengan memilih Auto-assign IPv4 di konsol atau menjalankan perintah berikut. Ganti *22XXXX01* dengan ID subnet Anda.

  ```
  aws ec2 modify-subnet-attribute --subnet-id subnet-22XXXX01 --map-public-ip-on-launch
  ```

- Untuk menjalankan Hive, Hue, atau Oozie di klaster EMR Amazon dengan beberapa node utama, Anda harus membuat metastore eksternal. Untuk informasi selengkapnya, lihat Mengkonfigurasi metastore eksternal untuk Hive, Menggunakan Hue dengan basis data jarak jauh di Amazon RDS, atau Apache Ozie.
- Untuk menggunakan otentikasi Kerberos di klaster Anda, Anda harus mengkonfigurasi KDC eksternal. Untuk informasi selengkapnya, lihat Mengonfigurasi Kerberos di Amazon Amazon EMR.

Luncurkan Amazon EMR Cluster dengan beberapa node utama

Anda dapat meluncurkan klaster dengan beberapa node utama saat Anda menggunakan grup instans atau armada instance. Bila Anda menggunakan grup instance dengan beberapa node primer, Anda harus menentukan nilai hitungan instance `3` untuk grup instance node primer. Saat Anda menggunakan armada instance dengan beberapa node primer, Anda harus menentukan `TargetOnDemandCapacity` dari`3`, `TargetSpotCapacity` dari `0` untuk armada instance utama, dan `WeightedCapacity` `1` untuk setiap jenis instance yang Anda konfigurasikan untuk armada utama.

Contoh berikut menunjukkan cara meluncurkan cluster menggunakan AMI default atau AMI kustom dengan grup instans dan armada instance:

> **Note**
>
> Anda harus menentukan ID subnet saat meluncurkan klaster EMR Amazon dengan beberapa node utama menggunakan. AWS CLI Ganti *22XXXX01* dan *22XXXX02* dengan subnet ID Anda dalam contoh berikut.

Default AMI, instance groups

Example Contoh - Meluncurkan cluster grup instans EMR Amazon dengan beberapa node primer menggunakan AMI default

```
aws emr create-cluster \
--name "ha-cluster" \
--release-label emr-6.15.0 \
--instance-groups InstanceGroupType=MASTER,InstanceCount=3,InstanceType=m5.xlarge
 InstanceGroupType=CORE,InstanceCount=4,InstanceType=m5.xlarge \
--ec2-attributes
 KeyName=ec2_key_pair_name,InstanceProfile=EMR_EC2_DefaultRole,SubnetId=subnet-22XXXX01
 \
--service-role EMR_DefaultRole \
--applications Name=Hadoop Name=Spark
```

Default AMI, instance fleets

Example Contoh - Meluncurkan cluster armada instans EMR Amazon dengan beberapa node primer menggunakan AMI default

```
aws emr create-cluster \
--name "ha-cluster" \
--release-label emr-6.15.0 \
--instance-fleets '[
    {
        "InstanceFleetType": "MASTER",
        "TargetOnDemandCapacity": 3,
        "TargetSpotCapacity": 0,
        "LaunchSpecifications": {
            "OnDemandSpecification": {
                "AllocationStrategy": "lowest-price"
            }
        },
        "InstanceTypeConfigs": [
            {
                "WeightedCapacity": 1,
                "BidPriceAsPercentageOfOnDemandPrice": 100,
                "InstanceType": "m5.xlarge"
            },
            {
                "WeightedCapacity": 1,
                "BidPriceAsPercentageOfOnDemandPrice": 100,
                "InstanceType": "m5.2xlarge"
            },
            {
                "WeightedCapacity": 1,
                "BidPriceAsPercentageOfOnDemandPrice": 100,
                "InstanceType": "m5.4xlarge"
            }
        ],
        "Name": "Master - 1"
    },
    {
        "InstanceFleetType": "CORE",
        "TargetOnDemandCapacity": 5,
        "TargetSpotCapacity": 0,
        "LaunchSpecifications": {
            "OnDemandSpecification": {
                "AllocationStrategy": "lowest-price"
```

```
            }
        },
        "InstanceTypeConfigs": [
            {
                "WeightedCapacity": 1,
                "BidPriceAsPercentageOfOnDemandPrice": 100,
                "InstanceType": "m5.xlarge"
            },
            {
                "WeightedCapacity": 2,
                "BidPriceAsPercentageOfOnDemandPrice": 100,
                "InstanceType": "m5.2xlarge"
            },
            {
                "WeightedCapacity": 4,
                "BidPriceAsPercentageOfOnDemandPrice": 100,
                "InstanceType": "m5.4xlarge"
            }
        ],
        "Name": "Core - 2"
    }
]' \
 --ec2-attributes '{"InstanceProfile":"EMR_EC2_DefaultRole","SubnetIds":
["subnet-22XXXX01", "subnet-22XXXX02"]}' \
--service-role EMR_DefaultRole \
--applications Name=Hadoop Name=Spark
```

Custom AMI, instance groups

Example Contoh - Meluncurkan cluster grup instans EMR Amazon dengan beberapa node utama menggunakan AMI kustom

```
aws emr create-cluster \
--name "custom-ami-ha-cluster" \
--release-label emr-6.15.0 \
--instance-groups InstanceGroupType=MASTER,InstanceCount=3,InstanceType=m5.xlarge
 InstanceGroupType=CORE,InstanceCount=4,InstanceType=m5.xlarge \
--ec2-attributes
 KeyName=ec2_key_pair_name,InstanceProfile=EMR_EC2_DefaultRole,SubnetId=subnet-22XXXX01
 \
--service-role EMR_DefaultRole \
--applications Name=Hadoop Name=Spark \
--custom-ami-id ami-MyAmiID
```

Custom AMI, instance fleets

Example Contoh - Meluncurkan cluster armada instans EMR Amazon dengan beberapa node utama menggunakan AMI kustom

```
aws emr create-cluster \
--name "ha-cluster" \
--release-label emr-6.15.0 \
--instance-fleets '[
    {
        "InstanceFleetType": "MASTER",
        "TargetOnDemandCapacity": 3,
        "TargetSpotCapacity": 0,
        "LaunchSpecifications": {
            "OnDemandSpecification": {
                "AllocationStrategy": "lowest-price"
            }
        },
        "InstanceTypeConfigs": [
            {
                "WeightedCapacity": 1,
                "BidPriceAsPercentageOfOnDemandPrice": 100,
                "InstanceType": "m5.xlarge"
            },
            {
                "WeightedCapacity": 1,
                "BidPriceAsPercentageOfOnDemandPrice": 100,
                "InstanceType": "m5.2xlarge"
            },
            {
                "WeightedCapacity": 1,
                "BidPriceAsPercentageOfOnDemandPrice": 100,
                "InstanceType": "m5.4xlarge"
            }
        ],
        "Name": "Master - 1"
    },
    {
        "InstanceFleetType": "CORE",
        "TargetOnDemandCapacity": 5,
        "TargetSpotCapacity": 0,
        "LaunchSpecifications": {
            "OnDemandSpecification": {
                "AllocationStrategy": "lowest-price"
```

```
            }
        },
        "InstanceTypeConfigs": [
            {
                "WeightedCapacity": 1,
                "BidPriceAsPercentageOfOnDemandPrice": 100,
                "InstanceType": "m5.xlarge"
            },
            {
                "WeightedCapacity": 2,
                "BidPriceAsPercentageOfOnDemandPrice": 100,
                "InstanceType": "m5.2xlarge"
            },
            {
                "WeightedCapacity": 4,
                "BidPriceAsPercentageOfOnDemandPrice": 100,
                "InstanceType": "m5.4xlarge"
            }
        ],
        "Name": "Core - 2"
    }
]' \
--ec2-attributes '{"InstanceProfile":"EMR_EC2_DefaultRole","SubnetIds":
["subnet-22XXXX01", "subnet-22XXXX02"]}' \
--service-role EMR_DefaultRole \
--applications Name=Hadoop Name=Spark \
--custom-ami-id ami-MyAmiID
```

Mengakhiri Cluster EMR Amazon dengan beberapa node utama

Untuk mengakhiri klaster EMR Amazon dengan beberapa node utama, Anda harus menonaktifkan perlindungan terminasi sebelum mengakhiri klaster, seperti yang ditunjukkan contoh berikut. Ganti *j-3KVTXXXXXX7UG* dengan ID klaster Anda.

```
aws emr modify-cluster-attributes --cluster-id j-3KVTXXXXXX7UG --no-termination-
protected
aws emr terminate-clusters --cluster-id j-3KVTXXXXXX7UG
```

Integrasi Amazon EMR dengan grup penempatan EC2

Saat meluncurkan beberapa kluster simpul utama EMR Amazon di Amazon EC2, Anda memiliki opsi untuk menggunakan strategi grup penempatan untuk menentukan bagaimana Anda ingin instance node utama digunakan untuk melindungi dari kegagalan perangkat keras.

Strategi grup penempatan didukung dimulai dengan Amazon EMR versi 5.23.0 sebagai opsi untuk beberapa kluster simpul utama. Saat ini, hanya tipe node primer yang didukung oleh strategi grup penempatan, dan `SPREAD` strategi diterapkan pada node primer tersebut. `SPREAD`Strategi ini menempatkan sekelompok kecil instance di perangkat keras dasar yang terpisah untuk mencegah hilangnya beberapa node primer jika terjadi kegagalan perangkat keras. Perhatikan bahwa permintaan peluncuran instans dapat gagal jika perangkat keras unik tidak mencukupi untuk memenuhi permintaan tersebut. Untuk informasi lebih lanjut tentang strategi penempatan EC2 dan batasan, lihat Grup penempatan di Panduan Pengguna EC2 untuk Instans Linux.

Terdapat batas awal dari Amazon EC2 dari 500 klaster yang mendukung strategi grup penempatan yang mana dapat diluncurkan per AWS wilayah. Kontak AWS dukungan untuk meminta peningkatan jumlah grup penempatan diperbolehkan. Anda dapat mengidentifikasi grup penempatan EC2 yang dibuat Amazon EMR dengan melacak pasangan nilai kunci yang diasosiasikan Amazon EMR dengan strategi grup penempatan EMR Amazon. Untuk informasi selengkapnya tentang tag instans klaster EC2, lihat Melihat instans klaster di Amazon EC2.

Melampirkan kebijakan terkelola grup penempatan ke Amazon eMRole

Strategi grup penempatan memerlukan kebijakan terkelola yang disebut`AmazonElasticMapReducePlacementGroupPolicy`, yang memungkinkan Amazon EMR membuat, menghapus, dan menjelaskan grup penempatan di Amazon EC2. Anda harus melampirkan `AmazonElasticMapReducePlacementGroupPolicy` ke peran layanan untuk Amazon EMR sebelum meluncurkan klaster EMR Amazon dengan beberapa node utama.

Anda juga dapat melampirkan kebijakan `AmazonEMRServicePolicy_v2` terkelola ke peran layanan EMR Amazon, bukan kebijakan terkelola grup penempatan. `AmazonEMRServicePolicy_v2`memungkinkan akses yang sama ke grup penempatan di Amazon EC2 sebagai. `AmazonElasticMapReducePlacementGroupPolicy` Untuk informasi selengkapnya, lihat Peran layanan untuk Amazon EMR (peran EMR).

Kebijakan terkelola `AmazonElasticMapReducePlacementGroupPolicy` adalah teks JSON berikut yang dibuat dan dikelola oleh Amazon EMR.

> **Note**
>
> Karena kebijakan `AmazonElasticMapReducePlacementGroupPolicy` terkelola diperbarui secara otomatis, kebijakan yang ditampilkan di sini mungkin out-of-date. Penggunaan AWS Konsol manajemen untuk meninjau kebijakan saat ini.

```
{
   "Version":"2012-10-17",
   "Statement":[
      {
         "Resource":"*",
         "Effect":"Allow",
         "Action":[
            "ec2:DeletePlacementGroup",
            "ec2:DescribePlacementGroups"
         ]
      },
      {
         "Resource":"arn:aws:ec2:*:*:placement-group/pg-*",
         "Effect":"Allow",
         "Action":[
            "ec2:CreatePlacementGroup"
         ]
      }
   ]
}
```

Luncurkan klaster EMR Amazon dengan beberapa node utama menggunakan strategi grup penempatan

Untuk meluncurkan klaster EMR Amazon yang memiliki beberapa node utama dengan strategi grup penempatan, lampirkan kebijakan terkelola grup penempatan `AmazonElasticMapReducePlacementGroupPolicy` ke peran EMR Amazon. Untuk informasi selengkapnya, lihat Melampirkan kebijakan terkelola grup penempatan ke Amazon eMRole.

Setiap kali Anda menggunakan peran ini untuk memulai klaster EMR Amazon dengan beberapa node utama, Amazon EMR mencoba meluncurkan klaster dengan `SPREAD` strategi yang diterapkan pada node utamanya. Jika Anda menggunakan peran yang tidak memiliki kebijakan `AmazonElasticMapReducePlacementGroupPolicy` terkelola grup penempatan yang

dilampirkan padanya, Amazon EMR mencoba meluncurkan klaster EMR Amazon yang memiliki beberapa node utama tanpa strategi grup penempatan.

Jika Anda meluncurkan klaster EMR Amazon yang memiliki beberapa node utama dengan `placement-group-configs` parameter menggunakan Amazon EMRAPI atau CLI, Amazon EMR hanya akan meluncurkan klaster jika Amazon eMRole memiliki kebijakan terkelola grup penempatan yang dilampirkan. `AmazonElasticMapReducePlacementGroupPolicy` Jika Amazon eMRole tidak memiliki kebijakan yang dilampirkan, klaster EMR Amazon dengan beberapa node utama mulai gagal.

Amazon EMR API

Example Contoh - Gunakan strategi grup penempatan untuk meluncurkan cluster grup instans dengan beberapa node utama dari Amazon EMR API

Saat Anda menggunakan RunJobFlow tindakan untuk membuat klaster EMR Amazon dengan beberapa node utama, setel `PlacementGroupConfigs` properti ke yang berikut. Saat ini, `MASTER` peran instans secara otomatis menggunakan `SPREAD` sebagai strategi grup penempatan.

```
{
   "Name":"ha-cluster",
   "PlacementGroupConfigs":[
      {
         "InstanceRole":"MASTER"
      }
   ],
   "ReleaseLabel": emr-6.15.0,
   "Instances":{
      "ec2SubnetId":"subnet-22XXXX01",
      "ec2KeyName":"ec2_key_pair_name",
      "InstanceGroups":[
         {
            "InstanceCount":3,
            "InstanceRole":"MASTER",
            "InstanceType":"m5.xlarge"
         },
         {
            "InstanceCount":4,
            "InstanceRole":"CORE",
            "InstanceType":"m5.xlarge"
         }
      ]
```

```
   },
   "JobFlowRole":"EMR_EC2_DefaultRole",
   "ServiceRole":"EMR_DefaultRole"
}
```

- Ganti *ha-klaster* dengan nama klaster ketersediaan tinggi.
- Ganti *subnet-22XXXX01* dengan ID subnet Anda.
- Ganti *ec2_key_pair_name* dengan nama pasangan kunci EC2 Anda untuk klaster ini. Pasangan kunci EC2 bersifat opsional dan hanya diperlukan jika Anda ingin menggunakan SSH untuk mengakses klaster Anda.

AWS CLI

Example Contoh - Gunakan strategi grup penempatan untuk meluncurkan cluster armada instance dengan beberapa node utama dari AWS Command Line Interface

Saat Anda menggunakan RunJobFlow tindakan untuk membuat klaster EMR Amazon dengan beberapa node utama, setel `PlacementGroupConfigs` properti ke yang berikut. Saat ini, `MASTER` peran instans secara otomatis menggunakan `SPREAD` sebagai strategi grup penempatan.

```
aws emr create-cluster \
--name "ha-cluster" \
--placement-group-configs InstanceRole=MASTER \
--release-label emr-6.15.0 \
--instance-fleets '[
    {
        "InstanceFleetType": "MASTER",
        "TargetOnDemandCapacity": 3,
        "TargetSpotCapacity": 0,
        "LaunchSpecifications": {
            "OnDemandSpecification": {
                "AllocationStrategy": "lowest-price"
            }
        },
        "InstanceTypeConfigs": [
            {
                "WeightedCapacity": 1,
                "BidPriceAsPercentageOfOnDemandPrice": 100,
                "InstanceType": "m5.xlarge"
            },
            {
```

```
                "WeightedCapacity": 1,
                "BidPriceAsPercentageOfOnDemandPrice": 100,
                "InstanceType": "m5.2xlarge"
            },
            {
                "WeightedCapacity": 1,
                "BidPriceAsPercentageOfOnDemandPrice": 100,
                "InstanceType": "m5.4xlarge"
            }
        ],
        "Name": "Master - 1"
    },
    {
        "InstanceFleetType": "CORE",
        "TargetOnDemandCapacity": 5,
        "TargetSpotCapacity": 0,
        "LaunchSpecifications": {
            "OnDemandSpecification": {
                "AllocationStrategy": "lowest-price"
            }
        },
        "InstanceTypeConfigs": [
            {
                "WeightedCapacity": 1,
                "BidPriceAsPercentageOfOnDemandPrice": 100,
                "InstanceType": "m5.xlarge"
            },
            {
                "WeightedCapacity": 2,
                "BidPriceAsPercentageOfOnDemandPrice": 100,
                "InstanceType": "m5.2xlarge"
            },
            {
                "WeightedCapacity": 4,
                "BidPriceAsPercentageOfOnDemandPrice": 100,
                "InstanceType": "m5.4xlarge"
            }
        ],
        "Name": "Core - 2"
    }
]' \
--ec2-attributes '{
    "KeyName": "ec2_key_pair_name",
    "InstanceProfile": "EMR_EC2_DefaultRole",
```

```
    "SubnetIds": [
        "subnet-22XXXX01",
        "subnet-22XXXX02"
    ]
}' \
--service-role EMR_DefaultRole \
--applications Name=Hadoop Name=Spark
```

- Ganti *ha-klaster* dengan nama klaster ketersediaan tinggi.
- Ganti *ec2_key_pair_name* dengan nama pasangan kunci EC2 Anda untuk klaster ini. Pasangan kunci EC2 bersifat opsional dan hanya diperlukan jika Anda ingin menggunakan SSH untuk mengakses klaster Anda.
- *Ganti subnet-22xxxx01 dan subnet-22xxxx02 dengan ID subnet Anda.*

Luncurkan cluster dengan beberapa node primer tanpa strategi grup penempatan

Untuk cluster dengan beberapa node primer untuk meluncurkan node primer tanpa strategi grup penempatan, Anda perlu melakukan salah satu hal berikut:

- Hapus kebijakan terkelola grup penempatan `AmazonElasticMapReducePlacementGroupPolicy` dari Amazon eMRole, atau
- Luncurkan cluster dengan beberapa node primer dengan `placement-group-configs` parameter menggunakan Amazon EMRAPI atau CLI `NONE` memilih sebagai strategi grup penempatan.

Amazon EMR API

Example — Meluncurkan cluster dengan beberapa node primer tanpa strategi grup penempatan menggunakan Amazon EMRAPI.

Saat menggunakan RunJobFlow tindakan untuk membuat cluster dengan beberapa node primer, atur `PlacementGroupConfigs` properti ke yang berikut.

```
{
   "Name":"ha-cluster",
   "PlacementGroupConfigs":[
      {
         "InstanceRole":"MASTER",
         "PlacementStrategy":"NONE"
```

```
      }
   ],
   "ReleaseLabel":"emr-5.30.1",
   "Instances":{
      "ec2SubnetId":"subnet-22XXXX01",
      "ec2KeyName":"ec2_key_pair_name",
      "InstanceGroups":[
         {
            "InstanceCount":3,
            "InstanceRole":"MASTER",
            "InstanceType":"m5.xlarge"
         },
         {
            "InstanceCount":4,
            "InstanceRole":"CORE",
            "InstanceType":"m5.xlarge"
         }
      ]
   },
   "JobFlowRole":"EMR_EC2_DefaultRole",
   "ServiceRole":"EMR_DefaultRole"
}
```

- Ganti *ha-klaster* dengan nama klaster ketersediaan tinggi.
- Ganti *subnet-22XXXX01* dengan ID subnet Anda.
- Ganti *ec2_key_pair_name* dengan nama pasangan kunci EC2 Anda untuk klaster ini. Pasangan kunci EC2 bersifat opsional dan hanya diperlukan jika Anda ingin menggunakan SSH untuk mengakses klaster Anda.

Amazon EMR CLI

Example — Meluncurkan cluster dengan beberapa node primer tanpa strategi grup penempatan menggunakan Amazon EMRCLI.

Saat menggunakan RunJobFlow tindakan untuk membuat cluster dengan beberapa node primer, atur `PlacementGroupConfigs` properti ke yang berikut.

```
aws emr create-cluster \
--name "ha-cluster" \
--placement-group-configs InstanceRole=MASTER,PlacementStrategy=NONE \
--release-label emr-5.30.1 \
```

```
--instance-groups InstanceGroupType=MASTER,InstanceCount=3,InstanceType=m5.xlarge
 InstanceGroupType=CORE,InstanceCount=4,InstanceType=m5.xlarge \
--ec2-attributes
 KeyName=ec2_key_pair_name,InstanceProfile=EMR_EC2_DefaultRole,SubnetId=subnet-22XXXX01
 \
--service-role EMR_DefaultRole \
--applications Name=Hadoop Name=Spark
```

- Ganti *ha-klaster* dengan nama klaster ketersediaan tinggi.
- Ganti *subnet-22XXXX01* dengan ID subnet Anda.
- Ganti *ec2_key_pair_name* dengan nama pasangan kunci EC2 Anda untuk klaster ini. Pasangan kunci EC2 bersifat opsional dan hanya diperlukan jika Anda ingin menggunakan SSH untuk mengakses klaster Anda.

Memeriksa konfigurasi strategi grup penempatan yang dilampirkan ke cluster dengan beberapa node primer

Anda dapat menggunakan Amazon EMR describe cluster API untuk melihat konfigurasi strategi grup penempatan yang dilampirkan ke cluster dengan beberapa node utama.

Example

```
aws emr describe-cluster --cluster-id "j-xxxxx"
{
   "Cluster":{
      "Id":"j-xxxxx",
      ...
      ...
      "PlacementGroups":[
         {
            "InstanceRole":"MASTER",
            "PlacementStrategy":"SPREAD"
         }
      ]
   }
}
```

Pertimbangan dan praktik terbaik

Pertimbangkan hal berikut saat Anda membuat klaster EMR Amazon dengan beberapa node utama:

> **Important**
>
> Untuk meluncurkan kluster EMR ketersediaan tinggi dengan beberapa node utama, kami sangat menyarankan Anda menggunakan rilis EMR Amazon terbaru. Ini memastikan bahwa Anda mendapatkan tingkat ketahanan dan stabilitas tertinggi untuk cluster ketersediaan tinggi Anda.

- Ketersediaan tinggi untuk armada misalnya didukung dengan rilis Amazon EMR 5.36.1, 6.8.1, 6.9.1, 6.10.1, 6.11.1, 6.12.0, dan yang lebih tinggi. Misalnya grup, ketersediaan tinggi didukung dengan rilis Amazon EMR 5.23.0 dan yang lebih tinggi. Untuk mempelajari selengkapnya, lihat Tentang Rilis EMR Amazon.
- Pada klaster ketersediaan tinggi, Amazon EMR hanya mendukung peluncuran node primer dengan instans On Demand. Ini memastikan ketersediaan tertinggi untuk cluster Anda.
- Anda masih dapat menentukan beberapa tipe instans untuk armada primer tetapi semua node utama dari cluster ketersediaan tinggi diluncurkan dengan tipe instance yang sama, termasuk penggantian untuk node primer yang tidak sehat.
- Untuk melanjutkan operasi, klaster ketersediaan tinggi dengan beberapa node primer membutuhkan dua dari tiga node primer agar sehat. Akibatnya, jika ada dua node utama yang gagal secara bersamaan, cluster EMR Anda akan gagal.
- Semua cluster EMR, termasuk cluster ketersediaan tinggi, diluncurkan dalam satu Availability Zone. Oleh karena itu, mereka tidak dapat mentolerir kegagalan Availability Zone. Dalam kasus pemadaman Availability Zone, Anda kehilangan akses ke cluster.
- Amazon EMR tidak menjamin ketersediaan tinggi untuk aplikasi sumber terbuka selain yang ditentukan dalam aplikasi. Aplikasi yang didukung di Amazon EMR Cluster dengan beberapa node utama
- Di Amazon EMR merilis 5.23.0 hingga 5.30.1, hanya dua dari tiga node utama untuk cluster grup instance yang dijalankan. HDFS NameNode

Pertimbangan untuk mengkonfigurasi subnet:

- Cluster EMR Amazon dengan beberapa node primer hanya dapat berada di satu Availability Zone atau subnet. Amazon EMR tidak dapat mengganti node utama yang gagal jika subnet sepenuhnya digunakan atau kelebihan langganan jika terjadi failover. Untuk menghindari skenario ini, Anda

disarankan untuk mendedikasikan seluruh subnet ke Amazon EMRCluster. Selain itu, pastikan bahwa ada cukup alamat IP pribadi yang tersedia di subnet.

Pertimbangan untuk mengonfigurasi simpul inti:

- Untuk memastikan node inti juga sangat tersedia, kami sarankan Anda meluncurkan setidaknya empat node inti. Jika Anda memutuskan untuk meluncurkan cluster yang lebih kecil dengan tiga atau lebih sedikit node inti, setel `dfs.replication parameter` ke setidaknya 2 untuk HDFS agar memiliki replikasi DFS yang memadai. Untuk informasi selengkapnya, lihat Konfigurasi HDFS.

⚠ Warning

1. Pengaturan `dfs.replication` ke 1 pada cluster dengan kurang dari empat node dapat menyebabkan hilangnya data HDFS jika satu node turun. Kami menyarankan Anda menggunakan cluster dengan setidaknya empat node inti untuk beban kerja produksi.
2. Amazon EMR tidak akan mengizinkan cluster untuk menskalakan node inti di bawah ini. `dfs.replication` Misalnya, jika`dfs.replication = 2`, jumlah minimum node inti adalah 2.
3. Saat Anda menggunakan Penskalaan Terkelola, Penskalaan Otomatis, atau memilih untuk mengubah ukuran klaster secara manual, sebaiknya atur `dfs.replication` ke 2 atau lebih tinggi.

Pertimbangan untuk Mengatur Alarm pada Metrik:

- Amazon EMR tidak menyediakan metrik khusus aplikasi tentang HDFS atau YARN. Kami berkomentar bahwa Anda mengatur alarm untuk memantau jumlah instance node utama. Konfigurasikan alarm menggunakan CloudWatch metrik Amazon berikut:`MultiMasterInstanceGroupNodesRunning`,`MultiMasterInstanceGroupNodesRunningP` atau. `MultiMasterInstanceGroupNodesRequested` CloudWatch akan memberi tahu Anda jika terjadi kegagalan dan penggantian simpul primer.
  - Jika `MultiMasterInstanceGroupNodesRunningPercentage` lebih rendah dari 1,0 dan lebih besar dari 0,5, cluster mungkin telah kehilangan simpul utama. Dalam situasi ini, Amazon EMR mencoba mengganti simpul utama.

- Jika `MultiMasterInstanceGroupNodesRunningPercentage` turun di bawah 0,5, dua node utama mungkin gagal. Dalam situasi ini, kuorum hilang dan cluster tidak dapat dipulihkan. Anda harus secara manual memigrasikan data dari klaster ini.

Untuk informasi selengkapnya, lihat Mengatur alarm pada metrik.

Klaster EMR pada AWS Outposts

Dimulai dengan Amazon EMR versi 5.28.0, Anda dapat membuat dan menjalankan klaster EMR di AWS Outposts. AWS Outposts mengaktifkan layanan AWS asli, infrastruktur, dan model operasi di fasilitas on premise. Di lingkungan AWS Outposts, Anda dapat menggunakan AWS API, alat, dan infrastruktur yang sama dengan yang Anda gunakan di AWS Cloud. Amazon EMR pada AWS Outposts ideal untuk beban kerja dengan latensi rendah yang perlu dijalankan dekat dengan data dan aplikasi on premise. Untuk informasi selengkapnya tentang AWS Outposts, lihat AWS Outposts Panduan Pengguna.

Prasyarat

Berikut adalah prasyarat untuk menggunakan Amazon EMR pada AWS Outposts:

- Anda harus menginstal dan mengkonfigurasi AWS Outposts di pusat data on premise Anda.
- Pastikan Anda memiliki koneksi jaringan yang dapat diandalkan antara lingkungan Outpost Anda dan Wilayah AWS.
- Pastikan Anda memiliki kapasitas yang cukup untuk jenis instans yang didukung EMR tersedia di Outpost Anda.

Batasan

Berikut ini adalah batasan penggunaan Amazon EMR pada AWS Outposts:

- Instans Sesuai Permintaan adalah satu-satunya opsi yang didukung untuk instans Amazon EC2. Instans Spot tidak tersedia untuk Amazon EMR pada AWS Outposts.
- Jika Anda memerlukan volume penyimpanan Amazon EBS tambahan, hanya SSD Tujuan Umum (GP2) yang didukung.

- Bucket S3 yang menyimpan objek dalam Wilayah AWS yang Anda tentukan adalah satu-satunya opsi S3 yang didukung untuk Amazon EMR di Outposts. S3 di Outposts tidak didukung untuk Amazon EMR aktif. AWS Outposts
- Hanya jenis instans berikut yang didukung oleh Amazon EMR pada AWS Outposts:

Kelas instans	Tipe instans
Tujuan umum	`m5.xlarge |m5.2xlarge | m5.4xlarge |m5.12xlarge | m5.24xlarge |m5d.xlarge | m5d.2xlarge | m5d.4xlarge | m5d.12xlarge |m5d.24xlarge`
Dioptimalkan komputasi	`c5.xlarge |c5.2xlarge | c5.4xlarge |c5.18xlarge | c5d.xlarge |c5d.2xlarge | c5d.4xlarge | c5d.18xlarge`
Memori-dioptimalkan	`r5.xlarge |r5.2xlarge | r5.4xlarge |r5.12xlarge | r5d.xlarge |r5d.2xlarge | r5d.4xlarge | r5d.12xlarge | r5d.24xlarge`
Penyimpanan dioptimalkan	`i3en.xlarge |i3en.2xlarge | i3en.3xlarge | i3en.6xlarge | i3en.12xlarge |i3en.24xlarge`

Pertimbangan konektivitas jaringan

- Jika konektivitas jaringan antara Outpost Anda dan Wilayah AWS hilang, klaster Anda akan terus berjalan. Namun, Anda tidak dapat membuat klaster baru atau mengambil tindakan baru pada klaster yang ada sampai konektivitas dipulihkan. Dalam kasus kegagalan instans, instans tidak akan diganti secara otomatis. Selain itu, tindakan seperti menambahkan langkah ke cluster yang sedang berjalan, memeriksa status eksekusi langkah, dan mengirim CloudWatch metrik dan peristiwa akan ditunda.
- Kami merekomendasikan Anda untuk menyediakan konektivitas jaringan yang andal dan tersedia penuh antara Outpost Anda dan Wilayah AWS. Jika konektivitas jaringan antara Outpost Anda dan Wilayah AWS hilang selama lebih dari beberapa jam, klaster yang telah mengaktifkan perlindungan

penghentian akan terus berjalan, dan klaster yang telah menonaktifkan perlindungan penghentian dapat dihentikan.

- Jika konektivitas jaringan akan terpengaruh karena pemeliharaan rutin, sebaiknya aktifkan perlindungan penghentian secara proaktif. Secara lebih umum, gangguan konektivitas berarti bahwa setiap dependensi eksternal yang tidak bersifat lokal ke Outpost atau jaringan pelanggan tidak akan dapat diakses. Ini termasuk Amazon S3, DynamoDB yang digunakan dengan tampilan konsistensi EMRFS, dan Amazon RDS jika instans dalam wilayah digunakan untuk klaster EMR Amazon dengan beberapa node utama.

Membuat klaster Amazon EMR pada AWS Outposts

Membuat klaster Amazon EMR pada AWS Outposts mirip dengan pembuatan klaster Amazon EMR di AWS Cloud. Saat Anda membuat klaster Amazon EMR pada AWS Outposts, Anda harus menentukan subnet Amazon EC2 yang terkait dengan Outpost Anda.

Amazon VPC dapat menjangkau semua Availability Zone di Wilayah AWS. AWS Outposts adalah ekstensi dari Availability Zone, dan Anda dapat memperluas Amazon VPC di akun untuk menjangkau beberapa Availability Zone dan lokasi Outpost terkait. Saat Anda mengonfigurasi Outpost, Anda mengaitkan subnet dengannya untuk memperluas lingkungan VPC Regional Anda ke fasilitas on premise. Instans outpost dan layanan terkait muncul sebagai bagian dari VPC Regional Anda, mirip dengan Availability Zone dan subnet terkait. Untuk informasi selengkapnya, lihat AWS Outposts Panduan Pengguna.

Konsol

Untuk membuat klaster Amazon EMR baru di AWS Outposts dengan AWS Management Console, tentukan subnet Amazon EC2 yang ditautkan dengan Outpost Anda.

> **Note**
>
> Kami telah mendesain ulang konsol EMR Amazon agar lebih mudah digunakan. Lihat Apa yang baru dengan konsol? untuk mempelajari tentang perbedaan antara pengalaman konsol lama dan baru.

New console

Untuk membuat cluster AWS Outposts dengan konsol baru

1. Masuk keAWS Management Console, dan buka konsol EMR Amazon di https://console.aws.amazon.com/emr.
2. Di bawah EMR pada EC2 di panel navigasi kiri, pilih Clusters, lalu pilih Create cluster.
3. Di bawah Konfigurasi klaster, pilih Grup instans atau Armada instans. Kemudian, pilih jenis instans dari menu tarik-turun Pilih jenis instans EC2 atau pilih Tindakan dan pilih Tambahkan volume EBS. Amazon EMR on AWS Outposts mendukung volume dan jenis instans Amazon EBS terbatas.
4. Di bawah Networking, pilih subnet EC2 dengan Outpost ID dalam format ini: op-123456789.
5. Pilih opsi lain yang berlaku untuk cluster Anda.
6. Untuk meluncurkan klaster Anda, pilih Buat klaster.

Old console

Untuk membuat cluster AWS Outposts dengan konsol lama

1. Arahkan ke konsol EMR Amazon baru dan pilih Beralih ke konsol lama dari navigasi samping. Untuk informasi selengkapnya tentang apa yang diharapkan saat beralih ke konsol lama, lihat Menggunakan konsol lama.
2. Pilih Buat klaster.
3. Pilih Pergi ke opsi lanjutan.
4. Di bawah Pengaturan perangkat lunak, untuk Rilis, pilih 5.28.0 atau yang lebih baru.
5. Di bawah Pengaturan perangkat keras, untuk Subnet EC2, pilih subnet EC2 dengan ID Outpost dalam format ini: op-123456789.
6. Pilih jenis instans atau tambahkan volume penyimpanan Amazon EBS untuk grup instans seragam atau armada instans. Volume dan tipe instans Amazon EBS terbatas didukung untuk Amazon EMR pada AWS Outposts.

CLI

Untuk membuat cluster AWS Outposts dengan AWS CLI

- Untuk membuat kluster EMR Amazon baruAWS CLI, tentukan subnet EC2 yang terkait dengan Outpost Anda, seperti pada contoh berikut. AWS Outposts Ganti *Subnet-22xxxx01* dengan ID subnet EC2 Anda sendiri.

```
aws emr create-cluster \
--name "Outpost cluster" \
--release-label emr-7.0.0 \
--applications Name=Spark \
--ec2-attributes KeyName=myKey SubnetId=subnet-22XXXX01 \
--instance-type m5.xlarge --instance-count 3 --use-default-roles
```

Klaster EMR di AWS Local Zones

Dimulai dengan Amazon EMR versi 5.28.0, Anda dapat membuat dan menjalankan klaster Amazon EMR pada subnet Local Zones AWS sebagai ekstensi logis dari Wilayah AWS yang mendukung Local Zones. Local Zones mengaktifkan fitur Amazon EMR dan subset layanan AWS, seperti layanan komputasi dan penyimpanan, untuk ditempatkan lebih dekat dengan pengguna agar memberikan akses latensi sangat rendah ke aplikasi yang berjalan secara lokal. Untuk daftar Local Zones yang tersedia, lihat AWSLocal Zones. Untuk informasi tentang pengaksesan Local Zones AWS yang tersedia, lihat Wilayah, Availability Zone, dan Local Zones.

Tipe instans yang didukung

Jenis instans berikut tersedia untuk klaster Amazon EMR di Local Zones. Jenis instans yang tersedia berbeda-beda menurut Wilayah.

Kelas instans	Tipe instans
Tujuan umum	m5.xlarge | m5.2xlarge | m5.4xlarge | m5.12xlarge | m5.24xlarge | m5d.xlarge | m5d.2xlarge | m5d.4xlarge | m5d.12xlarge | m5d.24xlarge

Kelas instans	Tipe instans
Dioptimalkan komputasi	c5.xlarge | c5.2xlarge | c5.4xlarge | c5.9xlarge | c5.18xlarge | c5d.xlarge | c5d.2xlarge | c5d.4xlarge| c5d.9xlarge | c5d.18xlarge
Memori-dioptimalkan	r5.xlarge | r5.2xlarge | r5.4xlarge | r5.12xlarge | r5d.xlarge | r5d.2xlarge | r5d.4xlarge | r5d.12xlarge | r5d.24xlarge
Penyimpanan dioptimalkan	i3en.xlarge | i3en.2xlarge | i3en.3xlarge | i3en.6xlarge | i3en.12xlarge | i3en.24xlarge

Membuat klaster Amazon EMR di Local Zones

Membuat klaster Amazon EMR pada AWS Local Zones dengan meluncurkan klaster Amazon EMR ke subnet Amazon VPC yang terkait dengan Local Zona. Anda dapat mengakses klaster menggunakan nama Local Zones, Konsol us-west-2-lax-1a in the US West (Oregon).

Local Zones saat ini tidak mendukung Amazon EMR Notebook atau koneksi langsung ke Amazon EMR menggunakan antarmuka VPC endpoint (). AWS PrivateLink

> **Note**
>
> Kami telah mendesain ulang konsol EMR Amazon agar lebih mudah digunakan. Lihat Apa yang baru dengan konsol? untuk mempelajari tentang perbedaan antara pengalaman konsol lama dan baru.

New console

Untuk membuat cluster di Zona Lokal dengan konsol baru

1. Masuk keAWS Management Console, dan buka konsol EMR Amazon di https://console.aws.amazon.com/emr.
2. Di bawah EMR pada EC2 di panel navigasi kiri, pilih Clusters, lalu pilih Create cluster.
3. Di bawah Jaringan, pilih subnet EC2 dengan ID Zona Lokal dalam format ini: subnet 123abc | us-west-2-lax-1a.

4. Pilih jenis instans atau tambahkan volume penyimpanan Amazon EBS untuk grup instans seragam atau armada instans.
5. Pilih opsi lain yang berlaku untuk cluster Anda.
6. Untuk meluncurkan klaster Anda, pilih Buat klaster.

Old console

Untuk membuat cluster di Zona Lokal dengan konsol lama

1. Arahkan ke konsol EMR Amazon baru dan pilih Beralih ke konsol lama dari navigasi samping. Untuk informasi selengkapnya tentang apa yang diharapkan saat beralih ke konsol lama, lihat Menggunakan konsol lama.
2. Pilih Buat klaster.
3. Pilih Pergi ke opsi lanjutan.
4. Di bawah Pengaturan perangkat lunak, untuk Rilis, pilih 5.28.0 atau yang lebih baru.
5. Di bawah Pengaturan perangkat keras, untuk Subnet EC2, pilih subnet EC2 dengan ID Zona Lokal dalam format ini: subnet 123abc | us-west-2-lax-1a.
6. Tambahkan volume penyimpanan Amazon EBS untuk grup instans seragam atau armada instans lalu pilih tipe instans.

CLI

Untuk membuat cluster di Zona Lokal dengan AWS CLI

- Gunakan perintah create-cluster, bersama dengan SubnetId untuk Local Zone seperti yang ditunjukkan pada contoh berikut. Ganti subnet-22xxxx1234567 dengan Local Zone dan ganti opsi lain yang diperlukan. SubnetId Untuk informasi selengkapnya, lihat https://docs.aws.amazon.com/cli/latest/reference/emr/create-cluster.html.

```
aws emr create-cluster \
--name "Local Zones cluster" \
--release-label emr-5.29.0 \
--applications Name=Spark \
--ec2-attributes KeyName=myKey,SubnetId=subnet-22XXXX1234567 \
--instance-type m5.xlarge --instance-count 3 --use-default-roles
```

Konfigurasi Docker

Amazon EMR 6.x mendukung Hadoop 3, yang memungkinkan YARN NodeManager meluncurkan kontainer baik secara langsung di cluster EMR Amazon atau di dalam wadah Docker. kontainer Docker menyediakan lingkungan eksekusi kustom di mana kode aplikasi berjalan. Lingkungan eksekusi kustom diisolasi dari lingkungan eksekusi YARN NodeManager dan aplikasi lainnya.

Kontainer Docker dapat menyertakan pustaka khusus yang digunakan oleh aplikasi dan mereka dapat menyediakan berbagai versi alat dan pustaka asli, seperti R dan Python. Anda dapat menggunakan alat Docker yang sudah dikenal untuk menentukan pustaka dan dependensi waktu aktif untuk aplikasi Anda.

Klaster Amazon EMR 6.x dikonfigurasi secara default untuk mengizinkan aplikasi YARN, seperti Spark, berjalan menggunakan kontainer Docker. Untuk menyesuaikan konfigurasi kontainer Anda, edit opsi dukungan Docker yang ditetapkan dalam file `yarn-site.xml` dan `container-executor.cfg` yang tersedia di `/etc/hadoop/conf` direktori. Untuk rincian tentang setiap opsi konfigurasi dan bagaimana ia digunakan, lihat Meluncurkan aplikasi menggunakan kontainer Docker.

Anda dapat memilih untuk menggunakan Docker saat mengirimkan tugas. Gunakan variabel berikut untuk menentukan waktu aktif Docker dan gambar Docker.

- `YARN_CONTAINER_RUNTIME_TYPE=docker`
- `YARN_CONTAINER_RUNTIME_DOCKER_IMAGE={`*`DOCKER_IMAGE_NAME`*`}`

Saat Anda menggunakan kontainer Docker untuk menjalankan aplikasi YARN Anda, YARN mengunduh gambar Docker yang Anda tentukan saat mengirimkan tugas. Agar YARN dapat menyelesaikan gambar Docker ini, gambar tersebut harus dikonfigurasi dengan registri Docker. Opsi konfigurasi untuk registri Docker bergantung pada apakah Anda men-deploy klaster menggunakan subnet publik atau pribadi.

Registri Docker

Registri Docker adalah suatu sistem penyimpanan dan distribusi untuk gambar Docker. Untuk Amazon EMR, kami menyarankan Anda menggunakan Amazon ECR, yang merupakan registri kontainer Docker yang dikelola sepenuhnya yang memungkinkan Anda membuat gambar kustom Anda sendiri dan menghostingnya dalam arsitektur yang sangat tersedia dan dapat diskalakan.

Pertimbangan penyebaran

Registri Docker memerlukan akses jaringan dari setiap host di klaster. Ini karena setiap host mengunduh gambar dari registri Docker saat aplikasi YARN Anda berjalan di klaster. Persyaratan konektivitas jaringan ini dapat membatasi pilihan akan registri Docker Anda, bergantung pada apakah Anda men-deploy klaster Amazon EMR ke dalam subnet publik atau privat.

Subnet publik

Ketika kluster EMR digunakan di subnet publik, node yang menjalankan YARN NodeManager dapat langsung mengakses registri apa pun yang tersedia melalui internet.

Subnet pribadi

Ketika kluster EMR diterapkan di subnet pribadi, node yang menjalankan YARN NodeManager tidak memiliki akses langsung ke internet. Gambar Docker dapat di-host di Amazon ECR dan diakses melalui AWS PrivateLink.

Untuk informasi selengkapnya tentang cara menggunakan AWS PrivateLink untuk mengizinkan akses ke Amazon ECR dalam skenario subnet privat, lihat Menyiapkan AWS PrivateLink untuk Amazon ECS, dan Amazon ECR.

Mengkonfigurasi registri Docker

Untuk menggunakan registri Docker dengan Amazon EMR, Anda harus mengonfigurasi Docker agar memercayai registri tertentu yang ingin Anda gunakan untuk menyelesaikan gambar Docker. Registrasi kepercayaan default adalah lokal (pribadi) dan centos. Untuk menggunakan repositori publik lain atau Amazon ECR, Anda dapat mengubah pengaturan `docker.trusted.registries` di `/etc/hadoop/conf/container-executor.cfg` menggunakan EMR Classification API dengan kunci klasifikasi `container-executor`.

Contoh berikut menunjukkan cara mengkonfigurasi klaster untuk mempercayai kedua repositori publik, bernama `your-public-repo`, dan titik akhir registri ECR,`123456789123.dkr.ecr.us-east-1.amazonaws.com`. Jika Anda menggunakan ECR, ganti titik akhir ini dengan titik akhir ECR khusus Anda.

```
[
  {
    "Classification": "container-executor",
    "Configurations": [
        {
            "Classification": "docker",
            "Properties": {
```

```
                "docker.trusted.registries": "local,centos,your-public-
repo,123456789123.dkr.ecr.us-east-1.amazonaws.com",
                "docker.privileged-containers.registries": "local,centos,your-public-
repo,123456789123.dkr.ecr.us-east-1.amazonaws.com"
            }
        }
    ]
  }
]
```

Untuk meluncurkan klaster Amazon EMR 6.0.0 dengan konfigurasi ini menggunakan AWS Command Line Interface (AWS CLI), buat file bernama `container-executor.json` dengan isi konfigurasi JSON ontainer-executor sebelumnya. Kemudian, gunakan perintah berikut untuk meluncurkan klaster.

```
export KEYPAIR=<Name of your Amazon EC2 key-pair>
export SUBNET_ID=<ID of the subnet to which to deploy the cluster>
export INSTANCE_TYPE=<Name of the instance type to use>
export REGION=<Region to which to deploy the cluster>

aws emr create-cluster \
    --name "EMR-6.0.0" \
    --region $REGION \
    --release-label emr-6.0.0 \
    --applications Name=Hadoop Name=Spark \
    --service-role EMR_DefaultRole \
    --ec2-attributes KeyName=$KEYPAIR,InstanceProfile=EMR_EC2_DefaultRole,SubnetId=
$SUBNET_ID \
    --instance-groups InstanceGroupType=MASTER,InstanceCount=1,InstanceType=
$INSTANCE_TYPE InstanceGroupType=CORE,InstanceCount=2,InstanceType=$INSTANCE_TYPE \
    --configuration file://container-executor.json
```

Mengonfigurasi YARN untuk mengakses Amazon ECR di EMR 6.0.0 dan yang lebih lama

Jika Anda baru mengenal Amazon ECR, ikuti petunjuk di Memulai dengan Amazon ECR dan memverifikasi bahwa Anda memiliki akses ke Amazon ECR dari setiap instans di klaster Amazon EMR Anda.

Pada EMR 6.0.0 dan yang lebih lama, untuk mengakses Amazon ECR menggunakan perintah Docker, Anda harus membuat kredensial terlebih dahulu. Untuk memverifikasi bahwa

YARN dapat mengakses gambar dari Amazon ECR, gunakan variabel lingkungan kontainer `YARN_CONTAINER_RUNTIME_DOCKER_CLIENT_CONFIG` untuk meneruskan referensi ke kredensials yang Anda buat.

Jalankan perintah berikut pada salah satu simpul inti untuk mendapatkan baris login untuk akun ECR Anda.

```
aws ecr get-login --region us-east-1 --no-include-email
```

Perintah `get-login` menghasilkan perintah CLI Docker yang benar untuk dijalankan dalam pembuatan kredensial. Salin dan jalankan output dari `get-login`.

```
sudo docker login -u AWS -p <password> https://<account-id>.dkr.ecr.us-
east-1.amazonaws.com
```

Perintah ini menghasilkan file `config.json` dalam folder `/root/.docker`. Salin file ini ke HDFS sehingga tugas yang dikirimkan ke klaster dapat menggunakannya untuk mengautentikasi ke Amazon ECR.

Jalankan perintah di bawah ini untuk menyalin file `config.json` ke direktori beranda Anda.

```
mkdir -p ~/.docker
sudo cp /root/.docker/config.json ~/.docker/config.json
sudo chmod 644 ~/.docker/config.json
```

Jalankan perintah di bawah ini untuk meletakkan config.json di HDFS sehingga dapat digunakan oleh tugas yang berjalan di klaster.

```
hadoop fs -put ~/.docker/config.json /user/hadoop/
```

YARN dapat mengakses ECR sebagai registri gambar Docker dan menarik kontainer selama eksekusi tugas.

Setelah mengonfigurasi registri Docker dan YARN, Anda dapat menjalankan aplikasi YARN menggunakan kontainer Docker. Untuk informasi selengkapnya, lihat Menjalankan aplikasi Spark dengan Docker menggunakan Amazon EMR 6.0.0.

Di EMR 6.1.0 dan yang lebih baru, Anda tidak harus mengatur autentikasi ke Amazon ECR secara manual. Jika registri Amazon ECR terdeteksi di kunci klasifikasi `container-executor`, fitur

autentikasi otomatis Amazon ECR akan diaktifkan, dan YARN menangani proses autentikasi saat Anda mengirimkan tugas Spark dengan gambar ECR. Anda dapat mengonfirmasi apakah autentikasi otomatis diaktifkan dengan memeriksa `yarn.nodemanager.runtime.linux.docker.ecr-auto-authentication.enabled` di situs yarn. Autentikasi otomatis diaktifkan dan pengaturan autentikasi YARN diatur ke `true` jika `docker.trusted.registries` mengandungi URL registri ECR.

Prasyarat untuk menggunakan otentikasi otomatis ke Amazon ECR

- EMR versi 6.1.0 atau lebih baru
- Registri ECR yang termasuk dalam konfigurasi berada di Wilayah yang sama dengan klaster
- IAM role dengan izin untuk mendapatkan token otorisasi dan menarik citra apa pun

Lihat Menyiapkan Amazon ECR untuk informasi lebih lanjut.

Cara mengaktifkan otentikasi otomatis

Ikuti Mengkonfigurasi registri Docker untuk mengatur registri Amazon ECR sebagai registri terpercaya, dan pastikan repositori Amazon ECR dan klaster berada di Wilayah yang sama.

Untuk mengaktifkan fitur ini bahkan ketika registri ECR tidak diatur dalam registri terpercaya, gunakan klasifikasi konfigurasi ini untuk mengatur `yarn.nodemanager.runtime.linux.docker.ecr-auto-authentication.enabled` ke `true`.

Cara menonaktifkan otentikasi otomatis

Secara default, autentikasi otomatis dinonaktifkan jika tidak ada registri Amazon ECR yang terdeteksi di registri yang tepercaya.

Untuk menonaktifkan autentikasi otomatis, bahkan ketika registri Amazon ECR diatur di registri tepercaya, gunakan klasifikasi konfigurasi ini untuk mengatur `yarn.nodemanager.runtime.linux.docker.ecr-auto-authentication.enabled` ke `false`.

Cara memeriksa apakah otentikasi otomatis diaktifkan pada cluster

Pada simpul utama, gunakan editor teks seperti `vi` untuk meninjau isi file: `vi /etc/hadoop/conf.empty/yarn-site.xml`. Periksa nilai `yarn.nodemanager.runtime.linux.docker.ecr-auto-authentication.enabled`.

Pengakhiran kontrol klaster

Bagian ini menjelaskan opsi Anda untuk mematikan kluster EMR Amazon. Ini mencakup perlindungan penghentian otomatis dan penghentian, dan bagaimana mereka berinteraksi dengan fitur EMR Amazon lainnya.

Anda dapat mematikan kluster EMR Amazon dengan cara berikut:

- Penghentian setelah eksekusi langkah terakhir - Buat cluster sementara yang mati setelah semua langkah selesai.
- Penghentian otomatis (setelah idle) - Buat klaster dengan kebijakan penghentian otomatis yang dimatikan setelah waktu idle yang ditentukan. Untuk informasi selengkapnya, lihat Menggunakan kebijakan penghentian otomatis.
- Pengakhiran manual - Buat cluster yang berjalan lama yang terus berjalan hingga Anda menghentikannya dengan sengaja. Untuk informasi tentang cara mengakhiri klaster secara manual, lihat Mengakhiri suatu klaster.

Anda juga dapat mengatur perlindungan terminasi pada klaster untuk menghindari mematikan instans EC2 secara tidak sengaja atau kesalahan.

Saat Amazon EMR mematikan klaster Anda, semua instans Amazon EC2 di klaster akan mati. Data di penyimpanan instans dan volume EBS tidak lagi tersedia dan tidak dapat dipulihkan. Memahami dan mengelola pengakhiran klaster sangat penting untuk mengembangkan strategi dalam mengelola dan menyimpan data dengan menulis ke Amazon S3 dan menyeimbangkan biaya.

Topik

Mengkonfigurasi cluster untuk melanjutkan atau mengakhiri setelah eksekusi langkah

Topik ini menjelaskan perbedaan antara menggunakan cluster yang berjalan lama dan membuat klaster sementara yang mati setelah langkah terakhir berjalan. Ini juga mencakup cara mengkonfigurasi eksekusi langkah untuk cluster.

Buat cluster yang berjalan lama

Secara default, cluster yang Anda buat dengan konsol atau AWS CLI sudah berjalan lama. Cluster yang berjalan lama terus berjalan, menerima pekerjaan, dan menambah biaya sampai Anda mengambil tindakan untuk mematikannya.

Cluster yang berjalan lama efektif dalam situasi berikut:

- Saat Anda perlu melakukan kueri data secara interaktif atau otomatis.
- Ketika Anda perlu berinteraksi dengan aplikasi data besar yang dihosting di cluster secara berkelanjutan.
- Ketika Anda secara berkala memproses kumpulan data yang begitu besar atau lebih sering sehingga tidak efisien untuk meluncurkan cluster baru dan memuat data setiap kali.

Anda juga dapat mengatur perlindungan terminasi pada klaster yang berjalan lama untuk menghindari mematikan instans EC2 secara tidak sengaja atau kesalahan. Untuk informasi selengkapnya, lihat Menggunakan perlindungan pengakhiran.

> **Note**
>
> Amazon EMR secara otomatis mengaktifkan perlindungan terminasi untuk semua cluster dengan beberapa node utama, dan mengganti pengaturan eksekusi langkah apa pun yang Anda berikan saat membuat klaster. Anda dapat menonaktifkan perlindungan terminasi setelah cluster diluncurkan. Lihat Mengonfigurasi perlindungan pengakhiran untuk menjalankan klaster. Untuk mematikan klaster dengan beberapa node primer, Anda harus terlebih dahulu memodifikasi atribut cluster untuk menonaktifkan perlindungan terminasi. Untuk petunjuk, silakan lihat Mengakhiri Cluster EMR Amazon dengan beberapa node utama.

Konfigurasikan cluster untuk mengakhiri setelah eksekusi langkah

Saat Anda mengonfigurasi penghentian setelah eksekusi langkah, cluster dimulai, menjalankan tindakan bootstrap, dan kemudian menjalankan langkah-langkah yang Anda tentukan. Segera setelah langkah terakhir selesai, Amazon EMR menghentikan instans Amazon EC2 cluster. Cluster yang Anda luncurkan dengan Amazon EMR API memiliki eksekusi langkah yang diaktifkan secara default.

Pengakhiran setelah eksekusi langkah efektif untuk cluster yang melakukan tugas pemrosesan berkala, seperti menjalankan pemrosesan data harian. Eksekusi langkah juga membantu Anda memastikan bahwa Anda ditagih hanya untuk waktu yang diperlukan untuk memproses data Anda. Untuk informasi selengkapnya tentang langkah-langkahnya, lihat Kirim pekerjaan ke sebuah klaster.

> **Note**
>
> Kami telah mendesain ulang konsol EMR Amazon agar lebih mudah digunakan. Lihat Apa yang baru dengan konsol? untuk mempelajari tentang perbedaan antara pengalaman konsol lama dan baru.

New console

Untuk mengaktifkan eksekusi langkah dengan konsol baru

1. Masuk keAWS Management Console, dan buka konsol EMR Amazon di https://console.aws.amazon.com/emr.
2. Di bawah EMR pada EC2 di panel navigasi kiri, pilih Clusters, lalu pilih Create cluster.
3. Di bawah Langkah, pilih Tambahkan langkah. Dalam Tambahkan langkah dialog, masukkan nilai bidang yang sesuai. Opsi akan berbeda tergantung pada tipe langkah. Untuk menambahkan langkah Anda dan keluar dari dialog, pilih Tambah langkah.
4. Di bawah Pengakhiran klaster, pilih kotak centang Hentikan klaster setelah langkah terakhir selesai.
5. Pilih opsi lain yang berlaku untuk cluster Anda.
6. Untuk meluncurkan klaster Anda, pilih Buat klaster.

Old console

Untuk mengaktifkan eksekusi langkah dengan konsol lama

1. Arahkan ke konsol EMR Amazon baru dan pilih Beralih ke konsol lama dari navigasi samping. Untuk informasi selengkapnya tentang apa yang diharapkan saat beralih ke konsol lama, lihat Menggunakan konsol lama.
2. Pilih Buat klaster.
3. Pilih Eksekusi langkah.

4. Pilih pengaturan lain yang sesuai untuk aplikasi Anda, lalu pilih Buat klaster.

AWS CLI

Untuk mengaktifkan eksekusi langkah dengan AWS CLI

- Tentukan parameter `--auto-terminate` saat Anda menggunakan perintah `create-cluster` untuk membuat klaster sementara.

  Contoh berikut menunjukkan bagaimana menggunakan `--auto-terminate` parameter. Anda dapat mengetik perintah berikut dan mengganti *myKey*dengan nama pasangan kunci EC2 anda.

  > **Note**
  >
  > Karakter lanjutan baris Linux (\) disertakan agar mudah dibaca. Karakter ini bisa dihapus atau digunakan dalam perintah Linux. Untuk Windows, hapus atau ganti dengan tanda sisipan (^).

```
aws emr create-cluster --name "Test cluster" --release-label emr-7.0.0 \
--applications Name=Hive Name=Pig --use-default-roles --ec2-attributes
 KeyName=myKey \
--steps Type=PIG,Name="Pig Program",ActionOnFailure=CONTINUE,\
Args=[-f,s3://mybucket/scripts/pigscript.pig,-p,\
INPUT=s3://mybucket/inputdata/,-p,OUTPUT=s3://mybucket/outputdata/,\
$INPUT=s3://mybucket/inputdata/,$OUTPUT=s3://mybucket/outputdata/]
--instance-type m5.xlarge --instance-count 3 --auto-terminate
```

API

Untuk mematikan eksekusi langkah dengan Amazon EMR API

- Saat Anda menggunakan RunJobFlowtindakan untuk membuat klaster, setel KeepJobFlowAliveWhenNoStepsproperti `ketrue`.

Menggunakan kebijakan penghentian otomatis

Kebijakan penghentian otomatis memungkinkan Anda mengatur pembersihan klaster tanpa perlu memantau dan menghentikan kluster yang tidak digunakan secara manual. Saat menambahkan kebijakan penghentian otomatis ke klaster, Anda menentukan jumlah waktu idle setelah klaster akan mati secara otomatis.

Bergantung pada versi rilis, Amazon EMR menggunakan kriteria yang berbeda untuk menandai cluster sebagai idle. Tabel berikut menguraikan bagaimana Amazon EMR menentukan kemalasan cluster.

Saat Anda menggunakan...	Sebuah cluster dianggap menganggur ketika...
Amazon EMR versi 5.34.0 dan yang lebih baru, dan 6.4.0 dan yang lebih baru	• Tidak ada aplikasi YARN aktif • Pemanfaatan HDFS di bawah 10% • Tidak ada notebook EMR aktif atau koneksi EMR Studio • Tidak ada antarmuka pengguna aplikasi on-cluster yang digunakan
Amazon EMR versi 5.30.0 - 5.33.0 dan 6.1.0 - 6.3.0	• Tidak ada aplikasi YARN aktif • Cluster tidak memiliki pekerjaan Spark aktif **Note** Amazon EMR menandai klaster sebagai idle dan dapat secara otomatis menghentikan klaster meskipun Anda memiliki kernel Python3 yang aktif. Ini karena menjalankan kernel Python3 tidak mengirimkan pekerjaan Spark di cluster. Untuk menggunakan penghenti

Saat Anda menggunakan...	Sebuah cluster dianggap menganggur ketika...
	an otomatis dengan kernel Python3, sebaiknya gunakan Amazon EMR versi 6.4.0 atau yang lebih baru.

> **Note**
>
> Amazon EMR versi 6.4.0 dan yang lebih baru mendukung file on-cluster untuk mendeteksi aktivitas pada node utama:. `/emr/metricscollector/isbusy` Saat Anda menggunakan klaster untuk menjalankan skrip shell atau aplikasi non-Yarn, Anda dapat menyentuh atau memperbarui secara berkala `isbusy` untuk memberi tahu Amazon EMR bahwa klaster tidak menganggur.

Anda dapat melampirkan kebijakan penghentian otomatis saat membuat klaster, atau menambahkan kebijakan ke klaster yang ada. Untuk mengubah atau menonaktifkan penghentian otomatis, Anda dapat memperbarui atau menghapus kebijakan.

Pertimbangan-pertimbangan

Pertimbangkan fitur dan batasan berikut sebelum menggunakan kebijakan penghentian otomatis:

- Di Asia Pasifik (Jakarta), pemutusan otomatis EMR Amazon tersedia dengan Amazon EMR 6.14.0 dan lebih tinggi.
- Berikut iniWilayah AWS, penghentian otomatis EMR Amazon tersedia dengan Amazon EMR 5.30.0 dan 6.1.0 dan yang lebih tinggi:

  AS Timur (Virginia N. dan Ohio), AS Barat (Oregon dan California N.), Amerika Selatan (Sao Paulo), Eropa (Frankfurt, Irlandia, London, Milan, Paris, dan Stockholm), Kanada (Tengah), Asia Pasifik (Hong Kong, Mumbai, Seoul, Singapura, Sydney, dan Tokyo), Timur Tengah (Bahrain), Afrika (Cape Town), (AS-Timur), (AS-Barat)), China (Beijing) dioperasikan oleh Sinnet, China AWS GovCloud (Ningxia) yang dioperasikan oleh NWCD. AWS GovCloud
- Batas waktu idle default menjadi 60 menit (satu jam) ketika Anda tidak menentukan jumlah. Anda dapat menentukan batas waktu idle minimum satu menit, dan batas waktu idle maksimum 7 hari.

- Dengan Amazon EMR versi 6.4.0 dan yang lebih baru, penghentian otomatis diaktifkan secara default saat Anda membuat cluster baru dengan konsol Amazon EMR.
- Amazon EMR menerbitkan Amazon CloudWatch metrik resolusi tinggi saat Anda mengaktifkan penghentian otomatis untuk klaster. Anda dapat menggunakan metrik ini untuk melacak aktivitas klaster dan kemalasan. Untuk informasi selengkapnya, lihat Metrik kapasitas klaster.
- Pengakhiran otomatis tidak didukung saat Anda menggunakan aplikasi berbasis non-Yarn seperti Presto, Trino, atau HBase.
- Untuk menggunakan penghentian otomatis, proses kolektor metrik harus dapat terhubung ke titik akhir API publik untuk penghentian otomatis di API Gateway. Jika Anda menggunakan nama DNS pribadi denganAmazon Virtual Private Cloud, penghentian otomatis tidak akan berfungsi dengan baik. Untuk memastikan bahwa penghentian otomatis berfungsi, kami sarankan Anda mengambil salah satu tindakan berikut:
  - Hapus titik akhir VPC antarmuka API Gateway dari VPC Amazon Anda.
  - Ikuti petunjuk di Mengapa saya mendapatkan kesalahan HTTP 403 Forbidden saat menghubungkan ke API Gateway API saya dari VPC? untuk menonaktifkan pengaturan nama DNS pribadi.
  - Luncurkan cluster Anda di subnet pribadi sebagai gantinya. Untuk informasi lebih lanjut, lihat topik diSubnet privat.
- (EMR 5.30.0 dan yang lebih baru) Jika Anda menghapus aturan default Izinkan Semua keluar ke 0.0.0.0/ untuk grup keamanan utama, Anda harus menambahkan aturan yang memungkinkan konektivitas TCP keluar ke grup keamanan Anda untuk akses layanan pada port 9443. Grup keamanan Anda untuk akses layanan juga harus mengizinkan lalu lintas TCP masuk pada port 9443 dari grup keamanan utama. Untuk informasi selengkapnya tentang mengonfigurasi grup keamanan, lihat grup keamanan yang dikelola Amazon EMR untuk contoh utama (subnet pribadi).

Izin untuk menggunakan penghentian otomatis

Sebelum dapat menerapkan dan mengelola kebijakan penghentian otomatis untuk Amazon EMR, Anda harus melampirkan izin yang tercantum dalam contoh kebijakan izin IAM berikut ke sumber daya IAM yang mengelola kluster EMR Anda.

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

```
      "Action": [
        "elasticmapreduce:PutAutoTerminationPolicy",
        "elasticmapreduce:GetAutoTerminationPolicy",
        "elasticmapreduce:RemoveAutoTerminationPolicy"
      ],
      "Resource": "<your-resources>"
    }
}
```

Lampirkan, perbarui, atau hapus kebijakan penghentian otomatis

Bagian ini menyertakan petunjuk untuk membantu Anda melampirkan, memperbarui, atau menghapus kebijakan penghentian otomatis dari klaster EMR Amazon. Sebelum Anda bekerja dengan kebijakan penghentian otomatis, pastikan Anda memiliki izin IAM yang diperlukan. Lihat Izin untuk menggunakan penghentian otomatis.

> **Note**
>
> Kami telah mendesain ulang konsol EMR Amazon agar lebih mudah digunakan. Lihat Apa yang baru dengan konsol? untuk mempelajari tentang perbedaan antara pengalaman konsol lama dan baru.

New console

Untuk melampirkan kebijakan penghentian otomatis saat Anda membuat klaster dengan konsol baru

1. Masuk keAWS Management Console, dan buka konsol EMR Amazon di https://console.aws.amazon.com/emr.
2. Di bawah EMR pada EC2 di panel navigasi kiri, pilih Clusters, lalu pilih Create cluster.
3. Di bawah Pengakhiran klaster, pilih Hentikan klaster setelah waktu idle.
4. Tentukan jumlah jam dan menit idle yang dapat berlalu sebelum cluster berakhir secara otomatis. Waktu idle default adalah 1 jam.
5. Pilih opsi lain yang berlaku untuk cluster Anda.
6. Untuk meluncurkan klaster Anda, pilih Buat klaster.

Untuk melampirkan, memperbarui, atau menghapus kebijakan penghentian otomatis pada klaster yang sedang berjalan dengan konsol baru

1. Masuk keAWS Management Console, dan buka konsol EMR Amazon di https://console.aws.amazon.com/emr.
2. Di bawah EMR pada EC2 di panel navigasi kiri, pilih Cluster, dan pilih cluster yang ingin Anda perbarui.
3. Pada tab Properties pada halaman detail cluster, temukan Pengakhiran cluster dan pilih Edit.
4. Pilih atau hapus Aktifkan penghentian otomatis untuk mengaktifkan atau menonaktifkan fitur. Jika Anda mengaktifkan penghentian otomatis, tentukan jumlah jam dan menit idle yang dapat berlalu sebelum cluster dihentikan secara otomatis. Kemudian pilih Simpan perubahan untuk mengonfirmasi.

Old console

Untuk melampirkan kebijakan penghentian otomatis saat Anda membuat klaster dengan konsol lama

1. Arahkan ke konsol EMR Amazon baru dan pilih Beralih ke konsol lama dari navigasi samping. Untuk informasi selengkapnya tentang apa yang diharapkan saat beralih ke konsol lama, lihat Menggunakan konsol lama.
2. Pilih Buat klaster.
3. Di bawah Konfigurasi perangkat keras, pilih Pengakhiran otomatis.
4. Tentukan jumlah jam dan menit idle setelah itu cluster harus dihentikan secara otomatis. Waktu idle default adalah satu jam.
5. Pilih pengaturan lain yang sesuai untuk aplikasi Anda, lalu pilih Buat klaster.

Untuk melampirkan, memperbarui, atau menghapus kebijakan penghentian otomatis pada klaster yang sedang berjalan dengan konsol lama

1. Arahkan ke konsol EMR Amazon baru dan pilih Beralih ke konsol lama dari navigasi samping. Untuk informasi selengkapnya tentang apa yang diharapkan saat beralih ke konsol lama, lihat Menggunakan konsol lama.
2. Pilih Cluster dan pilih cluster yang ingin Anda perbarui.
3. Pilih tab Hardware pada halaman detail cluster.

4. Pilih atau hapus Aktifkan penghentian otomatis untuk mengaktifkan atau menonaktifkan fitur. Jika Anda mengaktifkan penghentian otomatis, tentukan jumlah jam dan menit idle setelah itu cluster harus dihentikan secara otomatis.

AWS CLI

Sebelum Anda mulai

Sebelum Anda bekerja dengan kebijakan penghentian otomatis, kami sarankan Anda memperbarui ke versi terbaru. AWS CLI Untuk petunjuk, lihat Menginstal, memperbarui, dan menghapus instalasi. AWS CLI

Untuk melampirkan atau memperbarui kebijakan penghentian otomatis menggunakan AWS CLI

- Anda dapat menggunakan `aws emr put-auto-termination-policy` perintah untuk melampirkan atau memperbarui kebijakan penghentian otomatis di klaster.

  Contoh berikut menentukan 3600 detik untuk. *`IdleTimeout`* Jika Anda tidak menentukan *`IdleTimeout`*, nilai defaultnya menjadi satu jam.

  ```
  aws emr put-auto-termination-policy \
  --cluster-id <your-cluster-id> \
  --auto-termination-policy IdleTimeout=3600
  ```

  > **Note**
  >
  > Karakter lanjutan baris Linux (\) disertakan agar mudah dibaca Karakter ini bisa dihapus atau digunakan dalam perintah Linux. Untuk Windows, hapus atau ganti dengan tanda sisipan (^).

  Anda juga dapat menentukan nilai `--auto-termination-policy` saat Anda menggunakan `aws emr create-cluster` perintah. Untuk informasi selengkapnya tentang penggunaan perintah EMR Amazon diAWS CLI, lihat Referensi AWS CLIPerintah.

Untuk menghapus kebijakan penghentian otomatis dengan AWS CLI

- Gunakan `aws emr remove-auto-termination-policy` perintah untuk menghapus kebijakan penghentian otomatis dari klaster. Untuk informasi selengkapnya tentang penggunaan perintah EMR Amazon diAWS CLI, lihat Referensi AWS CLIPerintah.

```
aws emr remove-auto-termination-policy --cluster-id <your-cluster-id>
```

Menggunakan perlindungan pengakhiran

Saat perlindungan pengakhiran diaktifkan pada klaster yang berjalan lama, Anda masih dapat mengakhiri klaster, tetapi Anda harus secara gamblang menghapus perlindungan pengakhiran dari klaster terlebih dahulu. Ini membantu memastikan bahwa instans EC2 tidak dimatikan karena kecelakaan atau kesalahan. Perlindungan pengakhiran sangat berguna jika klaster Anda mungkin memiliki data yang disimpan di disk lokal yang perlu Anda pulihkan sebelum instans diakhiri. Anda dapat mengaktifkan perlindungan pengakhiran saat membuat klaster, dan Anda dapat mengubah pengaturan pada klaster yang sedang berjalan.

Dengan perlindungan penghentian yang diaktifkan, tindakan `TerminateJobFlows` di API Amazon EMR tidak bekerja. Pengguna tidak dapat mengakhiri klaster menggunakan API ini atau perintah `terminate-clusters` dari AWS CLI. API mengembalikan kesalahan, dan CLI keluar dengan kode pengembalian bukan nol. Saat Anda menggunakan konsol EMR Amazon untuk menghentikan klaster, Anda akan diminta langkah ekstra untuk menonaktifkan perlindungan penghentian.

> **Warning**
>
> Perlindungan pengakhiran tidak menjamin bahwa data akan disimpan jika terjadi kesalahan manusia atau suatu solusi—misalnya, jika perintah reboot dikeluarkan dari baris perintah saat terhubung ke instans menggunakan SSH, jika aplikasi atau skrip berjalan pada instans yang mengeluarkan perintah mulai ulang, atau jika Amazon EC2 atau Amazon EMR API digunakan untuk menonaktifkan perlindungan pengakhiran. Bahkan dengan perlindungan pengakhiran yang diaktifkan, data yang disimpan di penyimpanan instans, termasuk data HDFS, dapat hilang. Tulis keluaran data ke lokasi Amazon S3 dan buat strategi pencadangan yang sesuai untuk kebutuhan kelangsungan bisnis Anda.

Pengakhiran penghentian tidak memengaruhi kemampuan Anda untuk menskalakan sumber daya klaster menggunakan salah satu tindakan berikut:

- Mengubah ukuran cluster secara manual dengan AWS Management Console atauAWS CLI. Untuk informasi selengkapnya, lihat Secara manual mengubah ukuran klaster berjalan.
- Menghapus instance dari grup instans inti atau tugas menggunakan kebijakan penskalaan kedalam dengan penskalaan otomatis. Untuk informasi selengkapnya, lihat Menggunakan penskalaan otomatis dengan kebijakan kustom untuk grup instans .
- Menghapus instans dari armada instans dengan mengurangi kapasitas target. Untuk informasi selengkapnya, lihat Opsi armada instans.

Perlindungan pengakhiran dan Amazon EC2

Cluster Amazon EMR dengan perlindungan pengakhiran yang diaktifkan memiliki atribut `disableAPITermination` yang disetel untuk semua instans Amazon EC2 di klaster. Jika permintaan pengakhiran berasal dari Amazon EMR, kemudian pengaturan Amazon EMR dan Amazon EC2 untuk konflik instans, pengaturan Amazon EMR menggantikan pengaturan Amazon EC2. Misalnya, jika Anda menggunakan konsol Amazon EC2 untuk mengaktifkan perlindungan pengakhiran pada instans Amazon EC2 di klaster yang perlindungan pengakhirannya dinonaktifkan, saat Anda menggunakan konsol Amazon EMR, perintah AWS CLI untuk Amazon EMR, atau API Amazon EMR untuk mengakhiri klaster, Amazon EMR menyetel `DisableApiTermination` ke `false` dan mengakhiri instans bersama dengan instans lainnya.

> **Important**
>
> Jika instans dibuat sebagai bagian dari klaster Amazon EMR yang menggunakan perlindungan pengakhiran, dan API Amazon EC2 atau perintah AWS CLI digunakan untuk memodifikasi instans sehingga `DisableApiTermination` menjadi `false`, lalu API Amazon EC2 atau perintah AWS CLI menjalankan tindakan `TerminateInstances`, maka instans Amazon EC2 akan berakhir.

Perlindungan pengakhiran dan simpulYARN yang tidak sehat

Amazon EMR secara berkala memeriksa status Apache Hadoop YARN dari simpul yang berjalan pada instans Amazon EC2 inti dan tugas dalam sebuah klasster. Status kesehatan dilaporkan oleh layanan pemeriksa NodeManager kesehatan. Jika sebuah node melaporkanUNHEALTHY,

pengontrol instans EMR Amazon menolak mencantumkan node dan tidak mengalokasikan kontainer YARN ke sana sampai menjadi sehat kembali. Alasan umum untuk simpul yang tidak sehat adalah penggunaan disk yang melebihi 90%. Untuk informasi lebih lanjut tentang mengidentifikasi simpul yang tidak sehat dan pemulihan, lihat Kesalahan sumber daya.

Jika simpul tetap `UNHEALTHY` selama lebih dari 45 menit, Amazon EMR akan mengambil tindakan berikut berdasarkan status perlindungan pengakhiran.

Perlindungan pengakhiran	Hasil
Diaktifkan (Disarankan)	Instans inti Amazon EC2 tetap dalam status daftar penolakan dan terus diperhitungkan dalam kapasitas klaster. Anda dapat terhubung ke instans inti Amazon EC2 untuk konfigurasi dan pemulihan data, serta mengubah ukuran klaster Anda untuk menambah kapasitas. Untuk informasi selengkapnya, lihat Kesalahan sumber daya. Simpul tugas yang tidak sehat dikecualikan dari perlindungan pengakhiran dan akan diakhiri.
Nonaktif	Instans Amazon EC2 diakhiri. Amazon EMR menyediakan instans baru berdasarkan jumlah instans yang ditetapkan dalam grup instans atau kapasitas target untuk armada instans. Jika semua node inti `UNHEALTHY` selama lebih dari 45 menit, klaster berakhir, lalu melaporkan status `NO_SLAVES_LEFT` . **Important** Data HDFS mungkin hilang jika instans inti diakhiri karena status tidak sehat. Jika simpul menyimpan blok yang tidak direplikasi ke simpul lain, blok ini akan hilang, yang mana dapat menyebabk

Perlindungan pengakhiran	Hasil
	an hilangnya data. Kami menyarankan Anda menggunakan perlindungan pengakhiran sehingga Anda dapat terhubung ke instans dan memulihkan data jika diperlukan.

Perlindungan penghentian dan eksekusi langkah

Saat Anda mengaktifkan eksekusi langkah dan juga mengaktifkan perlindungan penghentian, Amazon EMR mengabaikan perlindungan penghentian.

Saat Anda mengirimkan langkah ke suatu klaster, Anda dapat mengatur properti `ActionOnFailure` untuk menentukan apa yang terjadi jika langkah tersebut tidak dapat menyelesaikan pelaksanaan karena mengalami kesalahan. Nilai yang mungkin untuk pengaturan ini adalah `TERMINATE_CLUSTER` (`TERMINATE_JOB_FLOW`dengan versi yang lebih lama), `CANCEL_AND_WAIT`, dan `CONTINUE`. Untuk informasi selengkapnya, lihat Kirim pekerjaan ke sebuah klaster.

Jika langkah gagal yang dikonfigurasi dengan `ActionOnFailure` set ke`CANCEL_AND_WAIT`, jika eksekusi langkah diaktifkan, cluster berakhir tanpa mengeksekusi langkah-langkah berikutnya.

Jika langkah yang mengalami kegagalan yang dikonfigurasi dengan set `ActionOnFailure` ke `TERMINATE_CLUSTER`, gunakan tabel pengaturan di bawah ini untuk menentukan hasilnya.

ActionOnFailure	Eksekusi langkah	Perlindungan pengakhiran	Hasil
`TERMINATE_CLUSTER`	Diaktifkan	Nonaktif	Klaster berakhir
	Diaktifkan	Diaktifkan	Klaster berakhir
	Nonaktif	Diaktifkan	Klaster berlanjut
	Nonaktif	Nonaktif	Klaster berakhir

Perlindungan pengakhiran dan Instans Spot

Perlindungan penghentian EMR Amazon tidak mencegah Instans Spot Amazon EC2 berakhir ketika harga Spot naik di atas harga Spot maksimum.

Mengonfigurasi perlindungan pengakhiran saat Anda meluncurkan klaster

Anda dapat mengaktifkan atau menonaktifkan perlindungan pengakhiran saat meluncurkan klaster menggunakan konsol, AWS CLI, atau API.

Pengaturan perlindungan pengakhiran default bergantung pada cara Anda meluncurkan klaster:

- Amazon EMR Console (baru) —Perlindungan Terminasi diaktifkan secara default.
- Amazon EMR Console (lama) Opsi Cepat — Perlindungan Penghentian dinonaktifkan secara default.
- Amazon EMR Console (lama) Opsi Lanjutan—Perlindungan Terminasi diaktifkan secara default.
- AWS CLI `aws emr create-cluster`—Perlindungan Pengakhiran dinonaktifkan kecuali `--termination-protected` ditentukan.
- Amazon EMR API RunJobFlowCommand—Perlindungan Penghentian dinonaktifkan kecuali nilai `TerminationProtected` boolean disetel ke. `true`

> **Note**
>
> Kami telah mendesain ulang konsol EMR Amazon agar lebih mudah digunakan. Lihat Apa yang baru dengan konsol? untuk mempelajari tentang perbedaan antara pengalaman konsol lama dan baru.

New console

Untuk mengaktifkan atau menonaktifkan perlindungan penghentian saat Anda membuat klaster dengan konsol baru

1. Masuk keAWS Management Console, dan buka konsol EMR Amazon di https://console.aws.amazon.com/emr.
2. Di bawah EMR pada EC2 di panel navigasi kiri, pilih Clusters, lalu pilih Create cluster.
3. Untuk versi rilis EMR, pilih emr-6.6.0 atau yang lebih baru.

4. Di bawah terminasi Cluster, pastikan bahwa perlindungan Use terminasi telah dipilih sebelumnya, atau hapus pilihan untuk mematikannya.
5. Pilih opsi lain yang berlaku untuk cluster Anda.
6. Untuk meluncurkan klaster Anda, pilih Buat klaster.

Old console

Untuk mengaktifkan atau menonaktifkan perlindungan terminasi saat Anda membuat klaster dengan konsol lama

1. Arahkan ke konsol EMR Amazon baru dan pilih Beralih ke konsol lama dari navigasi samping. Untuk informasi selengkapnya tentang apa yang diharapkan saat beralih ke konsol lama, lihat Menggunakan konsol lama.
2. Pilih Buat klaster.
3. Pilih Pergi ke opsi lanjutan.
4. Untuk Langkah 3: Pengaturan Klaster Umum, di bawah Opsi Umum pastikan Perlindungan pengakhiran dipilih untuk mengaktifkannya, atau hapus pilihan untuk menonaktifkannya.
5. Pilih pengaturan lain yang sesuai untuk aplikasi Anda, pilih Berikutnya, lalu selesaikan konfigurasi klaster Anda.

AWS CLI

Untuk mengaktifkan atau menonaktifkan perlindungan terminasi saat Anda membuat klaster menggunakan AWS CLI

- DenganAWS CLI, Anda dapat meluncurkan cluster dengan perlindungan terminasi diaktifkan dengan `create-cluster` perintah dengan `--termination-protected` parameter. Perlindungan pengakhiran dinonaktifkan secara default.

  Berikut adalah contoh membuat klaster dengan perlindungan pengakhiran yang diaktifkan:

  > **Note**
  >
  > Karakter lanjutan baris Linux (\) disertakan agar mudah dibaca. Karakter ini bisa dihapus atau digunakan dalam perintah Linux. Untuk Windows, hapus atau ganti dengan tanda pangkat (^).

```
aws emr create-cluster --name "TerminationProtectedCluster" --release-
label emr-7.0.0 \
--applications Name=Hadoop Name=Hive Name=Pig \
--use-default-roles --ec2-attributes KeyName=myKey --instance-type m5.xlarge \
--instance-count 3 --termination-protected
```

Untuk informasi selengkapnya tentang menggunakan perintah Amazon EMR dalam AWS CLI, lihat https://docs.aws.amazon.com/cli/latest/reference/emr.

Mengonfigurasi perlindungan pengakhiran untuk menjalankan klaster

Anda dapat mengonfigurasi perlindungan terminasi untuk cluster yang sedang berjalan dengan konsol atau fileAWS CLI.

> **Note**
>
> Kami telah mendesain ulang konsol EMR Amazon agar lebih mudah digunakan. Lihat Apa yang baru dengan konsol? untuk mempelajari tentang perbedaan antara pengalaman konsol lama dan baru.

New console

Untuk mengaktifkan atau menonaktifkan perlindungan terminasi untuk klaster yang sedang berjalan dengan konsol baru

1. Masuk keAWS Management Console, dan buka konsol EMR Amazon di https://console.aws.amazon.com/emr.
2. Di bawah EMR pada EC2 di panel navigasi kiri, pilih Cluster, dan pilih cluster yang ingin Anda perbarui.
3. Pada tab Properties pada halaman detail cluster, temukan Pengakhiran cluster dan pilih Edit.
4. Pilih atau kosongkan kotak centang Gunakan perlindungan penghentian untuk mengaktifkan atau menonaktifkan fitur. Kemudian pilih Simpan perubahan untuk mengonfirmasi.

Old console

Untuk mengaktifkan atau menonaktifkan perlindungan terminasi untuk cluster yang sedang berjalan dengan konsol lama

1. Arahkan ke konsol EMR Amazon baru dan pilih Beralih ke konsol lama dari navigasi samping. Untuk informasi selengkapnya tentang apa yang diharapkan saat beralih ke konsol lama, lihat Menggunakan konsol lama.
2. Pada halaman Klaster, pilih Nama klaster Anda.
3. Pada tab Ringkasan, untuk Perlindungan pengakhiran, pilih Ubah.
4. Untuk mengaktifkan perlindungan pengakhiran, pilih Hidup. Untuk menonaktifkan perlindungan pengakhiran, pilih Mati. Kemudian pilih tanda centang hijau untuk mengonfirmasi.

AWS CLI

Untuk mengaktifkan atau menonaktifkan perlindungan terminasi untuk klaster yang sedang berjalan menggunakan AWS CLI

- Untuk mengaktifkan perlindungan terminasi pada cluster yang sedang berjalan denganAWS CLI, gunakan `modify-cluster-attributes` perintah dengan `--termination-protected` parameter. Untuk menonaktifkannya, gunakan parameter `--no-termination-protected`.

  Berikut adalah contoh mengaktifkan perlindungan pengakhiran pada klaster dengan ID *j-3KVTXXXXXX7UG*:

  ```
  aws emr modify-cluster-attributes --cluster-id j-3KVTXXXXXX7UG --termination-protected
  ```

  Berikut adalah contoh menonaktifkan perlindungan pengakhiran pada klaster yang sama:

  ```
  aws emr modify-cluster-attributes --cluster-id j-3KVTXXXXXX7UG --no-termination-protected
  ```

Bekerja dengan AMI Amazon Linux di Amazon EMR

Gambar Mesin Amazon Amazon Linux (AMI)

Amazon EMR menggunakan Amazon Linux Amazon Machine Image (AMI) untuk menginisialisasi instans Amazon EC2 saat Anda membuat dan meluncurkan klaster. AMI berisi sistem operasi Amazon Linux, perangkat lunak lain, dan konfigurasi yang diperlukan setiap instans untuk menghosting aplikasi klaster Anda.

Secara default, saat Anda membuat klaster, Amazon EMR menggunakan AMI Amazon Linux default yang dibuat khusus untuk versi rilis Amazon EMR yang Anda gunakan. Untuk informasi selengkapnya tentang AMI Amazon Linux default, lihatMenggunakan AMI Amazon Linux default untuk Amazon EMR. Saat Anda menggunakan Amazon EMR 5.7.0 atau lebih tinggi, Anda dapat memilih untuk menentukan AMI Amazon Linux kustom alih-alih AMI Linux Amazon default untuk Amazon EMR. AMI kustom memungkinkan Anda mengenkripsi volume perangkat asal serta menyesuaikan aplikasi dan konfigurasi sebagai alternatif untuk menggunakan tindakan bootstrap. Anda dapat menentukan AMI kustom untuk setiap jenis instans dalam grup instans atau konfigurasi armada instans klaster EMR Amazon. Beberapa dukungan AMI kustom memberi Anda fleksibilitas untuk menggunakan lebih dari satu jenis arsitektur dalam sebuah cluster. Lihat Menggunakan AMI kustom.

Amazon EMR secara otomatis melampirkan volume SSD Tujuan Umum Amazon EBS sebagai perangkat asal untuk semua AMI. AMI yang didukung EBS meningkatkan kinerja. Untuk informasi selengkapnya tentang Amazon Linux AMI, lihat Amazon Machine Images (AMI). Untuk informasi selengkapnya tentang penyimpanan instans untuk instans Amazon EMR, lihat Penyimpanan instans.

Menggunakan AMI Amazon Linux default untuk Amazon EMR

Setiap versi rilis Amazon EMR menggunakan AMI Amazon Linux default untuk Amazon EMR kecuali Anda menentukan AMI khusus. Dimulai dengan rilis Amazon EMR 5.36 dan Amazon EMR 6.6, perilaku default untuk memperbarui Amazon Linux 2 (AL2) di AMI default Amazon EMR adalah menerapkan rilis AL2 terbaru secara otomatis untuk AMI EMR Amazon default.

Pembaruan Amazon Linux otomatis untuk rilis Amazon EMR

Saat Anda meluncurkan cluster dengan rilis patch terbaru Amazon EMR 7.0 atau lebih tinggi, 6.6 atau lebih tinggi, atau 5.36 atau lebih tinggi, Amazon EMR menggunakan rilis Amazon Linux terbaru untuk Amazon EMR AMI default. Sebagai contoh:

- Di mana ada `x.x.0` dan `x.x.1` rilis, `x.x.0` rilis berhenti mendapatkan pembaruan AMI saat `x.x.1` diluncurkan.
- Demikian pula, `x.x.1` berhenti mendapatkan pembaruan AMI saat `x.x.2` diluncurkan.
- Kemudian, ketika `x.y.0` rilis, `x.x.[latest]` terus menerima pembaruan AMI di sampingnya`x.y.[latest]`.

Untuk melihat apakah Anda menggunakan rilis patch terbaru yang dilambangkan dengan angka setelah titik desimal kedua () `6.8.1` untuk rilis EMR Amazon, lihat rilis yang tersedia di Panduan Rilis EMR Amazon, periksa dropdown rilis EMR Amazon saat Anda membuat cluster di konsol, atau gunakan tindakan API atau CLI. `ListReleaseLabels`list-release-labels Untuk mendapatkan pembaruan saat kami meluncurkan rilis EMR Amazon baru, berlangganan umpan RSS di Apa yang baru? halaman di Panduan Rilis.

Jika mau, Anda dapat memilih untuk meluncurkan cluster Anda dengan versi Amazon Linux yang pertama kali dikirimkan oleh rilis Amazon EMR. Untuk informasi tentang cara menentukan rilis Amazon Linux untuk klaster Anda, lihatMengubah rilis Amazon Linux saat Anda membuat klaster EMR.

Versi Amazon Linux default

Topik

- AMI default untuk Amazon EMR 6.6 dan lebih tinggi
- AMI standar untuk Amazon EMR 5.x

AMI default untuk Amazon EMR 6.6 dan lebih tinggi

Tabel berikut mencantumkan informasi Amazon Linux untuk versi patch terbaru dari Amazon EMR rilis 6.6.x dan yang lebih tinggi.

OsRelea Label (Versi AL)	Versi kernel AL	Tanggal yang tersedia	Wilayah AWS
2.0.2023 206.0	4.14.330	Desember 22, 2023	• `us-east-1` • `us-east-2`

OsRelea Label (Versi AL)	Versi kernel AL	Tanggal yang tersedia	Wilayah AWS
			• us-west-1 • us-west-2 • eu-north-1 • eu-west-1 • eu-west-2 • eu-west-3 • eu-central-1 • eu-central-2 (6,10 +) • eu-south-1 • eu-south-2 (6,10 +) • ap-east-1 • ap-south-1 • ap-south-2 (6,10 +) • ap-southeast-3 • ap-southeast-4 (6.8+ dan 5.36.1) • ap-northeast-1 • ap-northeast-2 • ap-northeast-3 • ap-southeast-1 • ap-southeast-2 • af-south-1 • sa-east-1 • me-central-1 (6,10 +)

OsRelea Label (Versi AL)	Versi kernel AL	Tanggal yang tersedia	Wilayah AWS
			• me-south-1 • ca-central-1 • il-central-1 (6.8+ dan 5.36.1) • us-gov-east-1 • us-gov-west-1 • cn-north-1 • cn-northeast-1

OsRelea Label (Versi AL)	Versi kernel AL	Tanggal yang tersedia	Wilayah AWS
2.0.2023 116.0	4.14.328	Desember 11, 2023	• us-east-1 • us-east-2 • us-west-1 • us-west-2 • eu-north-1 • eu-west-1 • eu-west-2 • eu-west-3 • eu-central-1 • eu-central-2 (6,10 +) • eu-south-1 • eu-south-2 (6,10 +) • ap-east-1 • ap-south-1 • ap-south-2 (6,10 +) • ap-southeast-3 • ap-southe ast-4 (6.8+ dan 5.36.1) • ap-northeast-1 • ap-northeast-2 • ap-northeast-3 • ap-southeast-1 • ap-southeast-2 • af-south-1 • sa-east-1

OsRelea Label (Versi AL)	Versi kernel AL	Tanggal yang tersedia	Wilayah AWS
			• me-central-1 (6,10 +) • me-south-1 • ca-central-1 • il-central-1 (6.8+ dan 5.36.1) • us-gov-east-1 • us-gov-west-1 • cn-north-1 • cn-northeast-1

OsRelea Label (Versi AL)	Versi kernel AL	Tanggal yang tersedia	Wilayah AWS
2.0.2023 101.0	4.14.327	17 November 2023	• us-east-1 • us-east-2 • us-west-1 • us-west-2 • eu-north-1 • eu-west-1 • eu-west-2 • eu-west-3 • eu-central-1 • eu-central-2 (6,10 +) • eu-south-1 • eu-south-2 (6,10 +) • ap-east-1 • ap-south-1 • ap-south-2 (6,10 +) • ap-southeast-3 • ap-southe ast-4 (6.8+ dan 5.36.1) • ap-northeast-1 • ap-northeast-2 • ap-northeast-3 • ap-southeast-1 • ap-southeast-2 • af-south-1 • sa-east-1

OsRelea Label (Versi AL)	Versi kernel AL	Tanggal yang tersedia	Wilayah AWS
			• me-central-1 (6,10 +) • me-south-1 • ca-central-1 • il-central-1 (6.8+ dan 5.36.1) • us-gov-east-1 • us-gov-west-1 • cn-north-1 • cn-northeast-1

OsRelea Label (Versi AL)	Versi kernel AL	Tanggal yang tersedia	Wilayah AWS
2.0.2023 020.1	4.14.326	November 07, 2023	• us-east-1 • us-east-2 • us-west-1 • us-west-2 • eu-north-1 • eu-west-1 • eu-west-2 • eu-west-3 • eu-central-1 • eu-central-2 (6,10 +) • eu-south-1 • eu-south-2 (6,10 +) • ap-east-1 • ap-south-1 • ap-south-2 (6,10 +) • ap-southeast-3 • ap-southe ast-4 (6.8+ dan 5.36.1) • ap-northeast-1 • ap-northeast-2 • ap-northeast-3 • ap-southeast-1 • ap-southeast-2 • af-south-1 • sa-east-1

OsRelea Label (Versi AL)	Versi kernel AL	Tanggal yang tersedia	Wilayah AWS
			• me-central-1 (6,10 +) • me-south-1 • ca-central-1 • il-central-1 (6.8+ dan 5.36.1) • us-gov-east-1 • us-gov-west-1 • cn-north-1 • cn-northeast-1

OsRelea Label (Versi AL)	Versi kernel AL	Tanggal yang tersedia	Wilayah AWS
2.0.2023 012.1	4.14.326	26 Oktober 2023	• us-east-1 • us-east-2 • us-west-1 • us-west-2 • eu-north-1 • eu-west-1 • eu-west-2 • eu-west-3 • eu-central-1 • eu-central-2 (6,10 +) • eu-south-1 • eu-south-2 (6,10 +) • ap-east-1 • ap-south-1 • ap-south-2 (6,10 +) • ap-southeast-3 • ap-southe ast-4 (6.8+ dan 5.36.1) • ap-northeast-1 • ap-northeast-2 • ap-northeast-3 • ap-southeast-1 • ap-southeast-2 • af-south-1 • sa-east-1

OsRelea Label (Versi AL)	Versi kernel AL	Tanggal yang tersedia	Wilayah AWS
			• me-central-1 (6,10 +) • me-south-1 • ca-central-1 • il-central-1 (6.8+ dan 5.36.1) • us-gov-east-1 • us-gov-west-1 • cn-north-1 • cn-northeast-1

OsRelea Label (Versi AL)	Versi kernel AL	Tanggal yang tersedia	Wilayah AWS
2.0.2023 926.0	4.14.322	19 Oktober 2023	• us-east-1 • us-east-2 • us-west-1 • us-west-2 • eu-north-1 • eu-west-1 • eu-west-2 • eu-west-3 • eu-central-1 • eu-central-2 (6,10 +) • eu-south-1 • eu-south-2 (6,10 +) • ap-east-1 • ap-south-1 • ap-south-2 (6,10 +) • ap-southeast-3 • ap-southe ast-4 (6.8+ dan 5.36.1) • ap-northeast-1 • ap-northeast-2 • ap-northeast-3 • ap-southeast-1 • ap-southeast-2 • af-south-1 • sa-east-1

OsRelea Label (Versi AL)	Versi kernel AL	Tanggal yang tersedia	Wilayah AWS
			• me-central-1 (6,10 +) • me-south-1 • ca-central-1 • il-central-1 (6.8+ dan 5.36.1) • us-gov-east-1 • us-gov-west-1 • cn-north-1 • cn-northeast-1

OsRelea Label (Versi AL)	Versi kernel AL	Tanggal yang tersedia	Wilayah AWS
2.0.2023 8906.0	4.14.322	Oktober 04, 2023	• us-east-1 • us-east-2 • us-west-1 • us-west-2 • eu-north-1 • eu-west-1 • eu-west-2 • eu-west-3 • eu-central-1 • eu-central-2 (6,10 +) • eu-south-1 • eu-south-2 (6,10 +) • ap-east-1 • ap-south-1 • ap-south-2 (6,10 +) • ap-southeast-3 • ap-southe ast-4 (6.8+ dan 5.36.1) • ap-northeast-1 • ap-northeast-2 • ap-northeast-3 • ap-southeast-1 • ap-southeast-2 • af-south-1 • sa-east-1

OsRelea Label (Versi AL)	Versi kernel AL	Tanggal yang tersedia	Wilayah AWS
			• me-central-1 (6,10 +) • me-south-1 • ca-central-1 • il-central-1 (6.9+ dan 5.36.1)

OsRelea Label (Versi AL)	Versi kernel AL	Tanggal yang tersedia	Wilayah AWS
2.0.2023 822.0	4.14.322	Agustus 30, 2023	• us-east-1 • us-east-2 • us-west-1 • us-west-2 • eu-north-1 • eu-west-1 • eu-west-2 • eu-west-3 • eu-central-1 • eu-central-2 (6,10 +) • eu-south-1 • eu-south-2 (6,10 +) • ap-east-1 • ap-south-1 • ap-south-2 (6,10 +) • ap-southeast-3 • ap-southe ast-4 (6.8+ dan 5.36.1) • ap-northeast-1 • ap-northeast-2 • ap-northeast-3 • ap-southeast-1 • ap-southeast-2 • af-south-1 • sa-east-1

OsRelea Label (Versi AL)	Versi kernel AL	Tanggal yang tersedia	Wilayah AWS
			• me-central-1 (6,10 +) • me-south-1 • ca-central-1 • il-central-1 (6.9+ dan 5.36.1)

OsRelea Label (Versi AL)	Versi kernel AL	Tanggal yang tersedia	Wilayah AWS
2.0.2023 808.0	4.14.320	24 Agustus 2023	• us-east-1 • us-east-2 • us-west-1 • us-west-2 • eu-north-1 • eu-west-1 • eu-west-2 • eu-west-3 • eu-central-1 • eu-central-2 (6,10 +) • eu-south-1 • eu-south-2 (6,10 +) • ap-east-1 • ap-south-1 • ap-south-2 (6,10 +) • ap-southeast-3 • ap-southe ast-4 (6.8+ dan 5.36.1) • ap-northeast-1 • ap-northeast-2 • ap-northeast-3 • ap-southeast-1 • ap-southeast-2 • af-south-1 • sa-east-1

OsRelea Label (Versi AL)	Versi kernel AL	Tanggal yang tersedia	Wilayah AWS
			• me-central-1 (6,10 +) • me-south-1 • ca-central-1 • il-central-1 (6.9+ dan 5.36.1) • us-gov-east-1 • us-gov-west-1 • cn-north-1 • cn-northeast-1

OsRelea Label (Versi AL)	Versi kernel AL	Tanggal yang tersedia	Wilayah AWS
2.0.2023 727.0	4.14.320	Agustus 14, 2023	• us-east-1 • us-east-2 • us-west-1 • us-west-2 • eu-north-1 • eu-west-1 • eu-west-2 • eu-west-3 • eu-central-1 • eu-central-2 (6,10 +) • eu-south-1 • eu-south-2 (6,10 +) • ap-east-1 • ap-south-1 • ap-south-2 (6,10 +) • ap-southeast-3 • ap-southe ast-4 (6.8+ dan 5.36.1) • ap-northeast-1 • ap-northeast-2 • ap-northeast-3 • ap-southeast-1 • ap-southeast-2 • af-south-1 • sa-east-1

OsRelea Label (Versi AL)	Versi kernel AL	Tanggal yang tersedia	Wilayah AWS
			• me-central-1 (6,10 +) • me-south-1 • ca-central-1 • il-central-1 (6.9+ dan 5.36.1)

OsRelea Label (Versi AL)	Versi kernel AL	Tanggal yang tersedia	Wilayah AWS
2.0.2023 719.0	4.14.320	2 Agustus 2023	• us-east-1 • us-east-2 • us-west-1 • us-west-2 • eu-north-1 • eu-west-1 • eu-west-2 • eu-west-3 • eu-central-1 • eu-central-2 (6,10 +) • eu-south-1 • eu-south-2 (6,10 +) • ap-east-1 • ap-south-1 • ap-south-2 (6,10 +) • ap-southeast-3 • ap-southe ast-4 (6.8+ dan 5.36.1) • ap-northeast-1 • ap-northeast-2 • ap-northeast-3 • ap-southeast-1 • ap-southeast-2 • af-south-1 • sa-east-1

OsRelea Label (Versi AL)	Versi kernel AL	Tanggal yang tersedia	Wilayah AWS
			• me-central-1 (6,10 +) • me-south-1 • ca-central-1 • il-central-1 (6.9+ dan 5.36.1)

OsRelea Label (Versi AL)	Versi kernel AL	Tanggal yang tersedia	Wilayah AWS
2.0.2023 628.0	4.14.318	Juli 12, 2023	• us-east-1 • us-east-2 • us-west-1 • us-west-2 • eu-north-1 • eu-west-1 • eu-west-2 • eu-west-3 • eu-central-1 • eu-central-2 (6,10 +) • eu-south-1 • eu-south-2 (6,10 +) • ap-east-1 • ap-south-1 • ap-south-2 (6,10 +) • ap-southeast-3 • ap-northeast-1 • ap-northeast-2 • ap-northeast-3 • ap-southeast-1 • ap-southeast-2 • af-south-1 • sa-east-1 • me-central-1 (6,10 +)

OsRelea Label (Versi AL)	Versi kernel AL	Tanggal yang tersedia	Wilayah AWS
			• me-south-1 • ca-central-1

OsRelea Label (Versi AL)	Versi kernel AL	Tanggal yang tersedia	Wilayah AWS
2.0.2023 612.0	4.14.314	Juni 23, 2023	• us-east-1 • us-east-2 • us-west-1 • us-west-2 • eu-north-1 • eu-west-1 • eu-west-2 • eu-west-3 • eu-central-1 • eu-central-2 (6,10 +) • eu-south-1 • eu-south-2 (6,10 +) • ap-east-1 • ap-south-1 • ap-south-2 (6,10 +) • ap-southeast-3 • ap-northeast-1 • ap-northeast-2 • ap-northeast-3 • ap-southeast-1 • ap-southeast-2 • af-south-1 • sa-east-1 • me-central-1 (6,10 +)

OsRelea Label (Versi AL)	Versi kernel AL	Tanggal yang tersedia	Wilayah AWS
			• me-south-1 • ca-central-1

OsRelea Label (Versi AL)	Versi kernel AL	Tanggal yang tersedia	Wilayah AWS
2.0.2023 504.1	4.14.313	16 Mei 2023	• us-east-1 • us-east-2 • us-west-1 • us-west-2 • eu-north-1 • eu-west-1 • eu-west-2 • eu-west-3 • eu-central-1 • eu-central-2 (6,10 +) • eu-south-1 • eu-south-2 (6,10 +) • ap-east-1 • ap-south-1 • ap-south-2 (6,10 +) • ap-southeast-3 • ap-northeast-1 • ap-northeast-2 • ap-northeast-3 • ap-southeast-1 • ap-southeast-2 • af-south-1 • sa-east-1 • me-central-1 • me-south-1

OsRelea Label (Versi AL)	Versi kernel AL	Tanggal yang tersedia	Wilayah AWS
			• ca-central-1

OsRelea Label (Versi AL)	Versi kernel AL	Tanggal yang tersedia	Wilayah AWS
2.0.2023 418.0	4.14.311	3 Mei 2023	• us-east-1 • us-east-2 • us-west-1 • us-west-2 • eu-north-1 • eu-west-1 • eu-west-2 • eu-west-3 • eu-central-1 • eu-central-2 (hanya 6.10) • eu-south-1 • eu-south-2 (hanya 6.10) • ap-east-1 • ap-south-1 • ap-south-2 (hanya 6.10) • ap-southeast-3 • ap-northeast-1 • ap-northeast-2 • ap-northeast-3 • ap-southeast-1 • ap-southeast-2 • af-south-1 • sa-east-1 • me-central-1

OsRelea Label (Versi AL)	Versi kernel AL	Tanggal yang tersedia	Wilayah AWS
			• me-south-1 • ca-central-1

OsRelea Label (Versi AL)	Versi kernel AL	Tanggal yang tersedia	Wilayah AWS
2.0.2023 404.1	4.14.311	April 18, 2023	• us-east-1 • us-east-2 • us-west-1 • us-west-2 • eu-north-1 • eu-west-1 • eu-west-2 • eu-west-3 • eu-central-1 • eu-central-2 • eu-south-1 • eu-south-2 • ap-east-1 • ap-south-1 • ap-south-2 • ap-southeast-3 • ap-northeast-1 • ap-northeast-2 • ap-northeast-3 • ap-southeast-1 • ap-southeast-2 • af-south-1 • sa-east-1 • me-central-1 • me-south-1 • ca-central-1

OsRelea Label (Versi AL)	Versi kernel AL	Tanggal yang tersedia	Wilayah AWS
2.0.2023 404.0	4.14.311	April 10, 2023	• `us-east-1` • `eu-west-3`

OsRelea Label (Versi AL)	Versi kernel AL	Tanggal yang tersedia	Wilayah AWS
2.0.2023 320.0	4.14.309	30 Maret 2023	• us-east-1 • us-east-2 • us-west-1 • us-west-2 • eu-north-1 • eu-west-1 • eu-west-2 • eu-west-3 • eu-central-1 • eu-central-2 • eu-south-1 • eu-south-2 • ap-east-1 • ap-south-1 • ap-south-2 • ap-southeast-3 • ap-northeast-1 • ap-northeast-2 • ap-northeast-3 • ap-southeast-1 • ap-southeast-2 • af-south-1 • sa-east-1 • me-central-1 • me-south-1 • ca-central-1

OsRelea Label (Versi AL)	Versi kernel AL	Tanggal yang tersedia	Wilayah AWS
2.0.2023 307.0	4.14.305	Maret 15, 2023	• us-east-1 • us-east-2 • us-west-1 • us-west-2 • eu-north-1 • eu-west-1 • eu-west-2 • eu-west-3 • eu-central-1 • eu-central-2 • eu-south-1 • eu-south-2 • ap-east-1 • ap-south-1 • ap-south-2 • ap-southeast-3 • ap-northeast-1 • ap-northeast-2 • ap-northeast-3 • ap-southeast-1 • ap-southeast-2 • af-south-1 • sa-east-1 • me-central-1 • me-south-1 • ca-central-1

OsRelea Label (Versi AL)	Versi kernel AL	Tanggal yang tersedia	Wilayah AWS
2.0.2023 207.0	4.14.304	3 Maret 2023	• us-east-1 • us-east-2 • us-west-1 • us-west-2 • eu-north-1 • eu-west-1 • eu-west-2 • eu-west-3 • eu-central-1 • eu-central-2 • eu-south-1 • eu-south-2 • ap-east-1 • ap-south-1 • ap-south-2 • ap-southeast-3 • ap-northeast-1 • ap-northeast-2 • ap-northeast-3 • ap-southeast-1 • ap-southeast-2 • af-south-1 • sa-east-1 • me-central-1 • me-south-1 • ca-central-1

OsRelea Label (Versi AL)	Versi kernel AL	Tanggal yang tersedia	Wilayah AWS
2.0.2023 119.1	4.14.301	9 Februari 2023	• us-east-1 • us-east-2 • us-west-1 • us-west-2 • eu-north-1 • eu-west-1 • eu-west-2 • eu-west-3 • eu-central-1 • eu-south-1 • ap-east-1 • ap-south-1 • ap-southeast-3 • ap-northeast-1 • ap-northeast-2 • ap-northeast-3 • ap-southeast-1 • ap-southeast-2 • af-south-1 • sa-east-1 • me-south-1 • ca-central-1

OsRelease Label (Versi AL)	Versi kernel AL	Tanggal yang tersedia	Wilayah AWS
2.0.20221210.1	4.14.301	Januari 12, 2023	• us-east-1 • us-east-2 • us-west-1 • us-west-2 • eu-north-1 • eu-west-1 • eu-west-2 • eu-west-3 • eu-central-1 • eu-south-1 • ap-east-1 • ap-south-1 • ap-southeast-3 • ap-northeast-1 • ap-northeast-2 • ap-northeast-3 • ap-southeast-1 • ap-southeast-2 • af-south-1 • sa-east-1 • me-south-1 • ca-central-1

OsRelea Label (Versi AL)	Versi kernel AL	Tanggal yang tersedia	Wilayah AWS
2.0.2022 103.3	4.14.296	Desember 5, 2022	• us-east-1 • us-east-2 • us-west-1 • us-west-2 • eu-north-1 • eu-west-1 • eu-west-2 • eu-west-3 • eu-central-1 • eu-south-1 • ap-east-1 • ap-south-1 • ap-southeast-3 • ap-northeast-1 • ap-northeast-2 • ap-northeast-3 • ap-southeast-1 • ap-southeast-2 • af-south-1 • sa-east-1 • me-south-1 • ca-central-1

OsRelease Label (Versi AL)	Versi kernel AL	Tanggal yang tersedia	Wilayah AWS
2.0.20220004.0	4.14.294	November 2, 2022	• us-east-1 • us-east-2 • us-west-1 • us-west-2 • eu-north-1 • eu-west-1 • eu-west-2 • eu-west-3 • eu-central-1 • eu-south-1 • ap-east-1 • ap-south-1 • ap-southeast-3 • ap-northeast-1 • ap-northeast-2 • ap-northeast-3 • ap-southeast-1 • ap-southeast-2 • af-south-1 • sa-east-1 • me-south-1 • ca-central-1

OsRelea Label (Versi AL)	Versi kernel AL	Tanggal yang tersedia	Wilayah AWS
2.0.2022 912.1	4.14.291	Oktober 7, 2022	• us-east-1 • us-east-2 • us-west-1 • us-west-2 • eu-north-1 • eu-west-1 • eu-west-2 • eu-west-3 • eu-central-1 • eu-south-1 • ap-east-1 • ap-south-1 • ap-southeast-3 • ap-northeast-1 • ap-northeast-2 • ap-northeast-3 • ap-southeast-1 • ap-southeast-2 • af-south-1 • sa-east-1 • me-south-1 • ca-central-1
2.0.2022 805.0	4.14.287	Agustus 30, 2022	• us-west-1

OsRelea Label (Versi AL)	Versi kernel AL	Tanggal yang tersedia	Wilayah AWS
2.0.2022 719.0	4.14.287	Agustus 10, 2022	• us-east-1 • us-east-2 • us-west-1 • us-west-2 • eu-north-1 • eu-west-1 • eu-west-2 • eu-west-3 • eu-central-1 • eu-south-1 • ap-east-1 • ap-south-1 • ap-southeast-3 • ap-northeast-1 • ap-northeast-2 • ap-northeast-3 • ap-southeast-1 • ap-southeast-2 • af-south-1 • sa-east-1 • me-south-1 • ca-central-1

OsRelea Label (Versi AL)	Versi kernel AL	Tanggal yang tersedia	Wilayah AWS
2.0.2022 426.0	4.14.281	Juni 10, 2022	• us-east-1 • us-east-2 • us-west-1 • us-west-2 • eu-north-1 • eu-west-1 • eu-west-2 • eu-west-3 • eu-central-1 • eu-south-1 • ap-east-1 • ap-south-1 • ap-southeast-3 • ap-northeast-1 • ap-northeast-2 • ap-northeast-3 • ap-southeast-1 • ap-southeast-2 • af-south-1 • sa-east-1 • me-south-1 • ca-central-1

OsRelea Label (Versi AL)	Versi kernel AL	Tanggal yang tersedia	Wilayah AWS
2.0.2022 406.1	4.14.275	Mei 2, 2022	• us-east-1 • us-east-2 • us-west-1 • us-west-2 • eu-north-1 • eu-west-1 • eu-west-2 • eu-west-3 • eu-central-1 • eu-south-1 • ap-east-1 • ap-south-1 • ap-southeast-3 • ap-northeast-1 • ap-northeast-2 • ap-northeast-3 • ap-southeast-1 • ap-southeast-2 • af-south-1 • sa-east-1 • me-south-1 • ca-central-1

AMI standar untuk Amazon EMR 5.x

Tabel berikut mencantumkan informasi Amazon Linux untuk versi patch terbaru Amazon EMR 5.x rilis 5.36 dan lebih tinggi.

OsRelease Label (Versi AL)	Versi kernel AL	Tanggal yang tersedia	Wilayah AWS
2.0.20230504.1	4.14.313	16 Mei 2023	• us-east-1 • us-east-2 • us-west-1 • us-west-2 • ca-central-1 • eu-north-1 • eu-west-1 • eu-west-2 • eu-west-3 • eu-central-1 • eu-south-1 • ap-east-1 • ap-south-1 • ap-southeast-3 • ap-northeast-1 • ap-northeast-2 • ap-northeast-3 • ap-southeast-1 • ap-southeast-2 • af-south-1 • sa-east-1 • me-south-1 • me-central-1
2.0.20230418.0	4.14.311	3 Mei 2023	• us-east-1 • us-east-2

OsRelea Label (Versi AL)	Versi kernel AL	Tanggal yang tersedia	Wilayah AWS
			• us-west-1 • us-west-2 • ca-central-1 • eu-north-1 • eu-west-1 • eu-west-2 • eu-west-3 • eu-central-1 • eu-south-1 • ap-east-1 • ap-south-1 • ap-southeast-3 • ap-northeast-1 • ap-northeast-2 • ap-northeast-3 • ap-southeast-1 • ap-southeast-2 • af-south-1 • sa-east-1 • me-south-1 • me-central-1

OsRelea Label (Versi AL)	Versi kernel AL	Tanggal yang tersedia	Wilayah AWS
2.0.2023 404.1	4.14.311	April 18, 2023	• us-east-1 • us-east-2 • us-west-1 • us-west-2 • ca-central-1 • eu-north-1 • eu-west-1 • eu-west-2 • eu-west-3 • eu-central-1 • eu-south-1 • ap-east-1 • ap-south-1 • ap-southeast-3 • ap-northeast-1 • ap-northeast-2 • ap-northeast-3 • ap-southeast-1 • ap-southeast-2 • af-south-1 • sa-east-1 • me-south-1
2.0.2023 404.0	4.14.311	April 10, 2023	• us-east-1 • eu-west-3

OsRelease Label (Versi AL)	Versi kernel AL	Tanggal yang tersedia	Wilayah AWS
2.0.20230320.0	4.14.309	30 Maret 2023	• us-east-1 • us-east-2 • us-west-1 • us-west-2 • ca-central-1 • eu-north-1 • eu-west-1 • eu-west-2 • eu-west-3 • eu-central-1 • eu-south-1 • ap-east-1 • ap-south-1 • ap-southeast-3 • ap-northeast-1 • ap-northeast-2 • ap-northeast-3 • ap-southeast-1 • ap-southeast-2 • af-south-1 • sa-east-1 • me-south-1

OsRelease Label (Versi AL)	Versi kernel AL	Tanggal yang tersedia	Wilayah AWS
2.0.20230307.0	4.14.305	Maret 15, 2023	• us-east-1 • us-east-2 • us-west-1 • us-west-2 • ca-central-1 • eu-north-1 • eu-west-1 • eu-west-2 • eu-west-3 • eu-central-1 • eu-south-1 • ap-east-1 • ap-south-1 • ap-southeast-3 • ap-northeast-1 • ap-northeast-2 • ap-northeast-3 • ap-southeast-1 • ap-southeast-2 • af-south-1 • sa-east-1 • me-south-1

OsRelea Label (Versi AL)	Versi kernel AL	Tanggal yang tersedia	Wilayah AWS
2.0.2023 207.0	4.14.304	3 Maret 2023	• us-east-1 • us-east-2 • us-west-1 • us-west-2 • eu-north-1 • eu-west-1 • eu-west-2 • eu-west-3 • eu-central-1 • eu-south-1 • ap-east-1 • ap-south-1 • ap-southeast-3 • ap-northeast-1 • ap-northeast-2 • ap-northeast-3 • ap-southeast-1 • ap-southeast-2 • af-south-1 • sa-east-1 • me-south-1 • ca-central-1

OsRelea Label (Versi AL)	Versi kernel AL	Tanggal yang tersedia	Wilayah AWS
2.0.2022 210.1	4.14.301	Januari 12, 2023	• us-east-1 • us-east-2 • us-west-1 • us-west-2 • eu-north-1 • eu-west-1 • eu-west-2 • eu-west-3 • eu-central-1 • eu-south-1 • ap-east-1 • ap-south-1 • ap-southeast-3 • ap-northeast-1 • ap-northeast-2 • ap-northeast-3 • ap-southeast-1 • ap-southeast-2 • af-south-1 • sa-east-1 • me-south-1 • ca-central-1

OsRelease Label (Versi AL)	Versi kernel AL	Tanggal yang tersedia	Wilayah AWS
2.0.20221103.3	4.14.296	Desember 5, 2022	• us-east-1 • us-east-2 • us-west-1 • us-west-2 • eu-north-1 • eu-west-1 • eu-west-2 • eu-west-3 • eu-central-1 • eu-south-1 • ap-east-1 • ap-south-1 • ap-southeast-3 • ap-northeast-1 • ap-northeast-2 • ap-northeast-3 • ap-southeast-1 • ap-southeast-2 • af-south-1 • sa-east-1 • me-south-1 • ca-central-1

OsRelea Label (Versi AL)	Versi kernel AL	Tanggal yang tersedia	Wilayah AWS
2.0.2022 004.0	4.14.294	November 2, 2022	• us-east-1 • us-east-2 • us-west-1 • us-west-2 • eu-north-1 • eu-west-1 • eu-west-2 • eu-west-3 • eu-central-1 • eu-south-1 • ap-east-1 • ap-south-1 • ap-southeast-3 • ap-northeast-1 • ap-northeast-2 • ap-northeast-3 • ap-southeast-1 • ap-southeast-2 • af-south-1 • sa-east-1 • me-south-1 • ca-central-1

OsRelease Label (Versi AL)	Versi kernel AL	Tanggal yang tersedia	Wilayah AWS
2.0.20220912.1	4.14.291	Oktober 7, 2022	• us-east-1 • us-east-2 • us-west-1 • us-west-2 • eu-north-1 • eu-west-1 • eu-west-2 • eu-west-3 • eu-central-1 • eu-south-1 • ap-east-1 • ap-south-1 • ap-southeast-3 • ap-northeast-1 • ap-northeast-2 • ap-northeast-3 • ap-southeast-1 • ap-southeast-2 • af-south-1 • sa-east-1 • me-south-1 • ca-central-1

OsRelea Label (Versi AL)	Versi kernel AL	Tanggal yang tersedia	Wilayah AWS
2.0.2022 719.0	4.14.287	Agustus 10, 2022	• us-west-1 • eu-west-3 • eu-north-1 • eu-central-1 • ap-south-1 • me-south-1

OsRelea Label (Versi AL)	Versi kernel AL	Tanggal yang tersedia	Wilayah AWS
2.0.2022 426.0	4.14.281	14 Juni 2022	• us-east-1 • us-east-2 • us-west-1 • us-west-2 • eu-north-1 • eu-west-1 • eu-west-2 • eu-west-3 • eu-central-1 • eu-south-1 • ap-east-1 • ap-south-1 • ap-southeast-3 • ap-northeast-1 • ap-northeast-2 • ap-northeast-3 • ap-southeast-1 • ap-southeast-2 • af-south-1 • sa-east-1 • me-south-1 • ca-central-1

Pertimbangan pembaruan perangkat lunak

Perhatikan perilaku pembaruan perangkat lunak default berikut:

Amazon EMR 7.x - Amazon Linux 2023

Amazon EMR merilis versi 7.0 dan lebih tinggi di Amazon Linux 2023 (AL2023). Perilaku default untuk AL2023 adalah mengunci AMI ke versi tertentu dari repositori perangkat lunak Amazon Linux. Oleh karena itu, pembaruan keamanan tidak diterapkan setiap kali Anda meluncurkan cluster. Sebagai gantinya, perilaku default untuk rilis Amazon EMR 7.x adalah menerapkan rilis AL2023 terbaru secara otomatis untuk AMI EMR Amazon default hanya saat Anda membuat cluster. Untuk menerima pembaruan keamanan terbaru, kami sarankan Anda membuat ulang cluster Anda secara berkala.

Amazon EMR 5.x dan 6.x - Amazon Linux dan Amazon Linux 2

Untuk rilis Amazon EMR yang lebih rendah dari 7.0, ketika instans Amazon EC2 melakukan booting untuk pertama kalinya dalam cluster yang didasarkan pada AMI Amazon Linux (AL) atau Amazon Linux 2 (AL2) default untuk Amazon EMR, ia memeriksa pembaruan perangkat lunak yang berlaku untuk versi rilis di repositori paket yang diaktifkan untuk AL dan Amazon EMR. Seperti instans AL dan AL2 lainnya, pembaruan keamanan penting dan penting dari repositori ini diinstal secara otomatis.

Perhatikan juga bahwa, dalam konfigurasi jaringan Anda, Anda harus mengizinkan jalan keluar HTTP dan HTTPS ke repositori Amazon Linux di Amazon S3. Jika tidak, pembaruan keamanan akan gagal. Untuk informasi selengkapnya, lihat Amazon Linux - Package repositori di Panduan Pengguna Amazon EC2 untuk Instans Linux. Secara default, paket perangkat lunak lain dan pembaruan kernel yang memerlukan reboot, termasuk NVIDIA dan CUDA, dikecualikan dari pengunduhan otomatis saat boot pertama.

Amazon EMR 5.35.0 dan lebih rendah, dan 6.5.0 dan lebih rendah - Amazon Linux AMI dikunci ke versi rilis Amazon EMR

Untuk Amazon EMR 5.35.0 dan yang lebih rendah, dan 6.5.0 dan yang lebih rendah, AMI default didasarkan pada AMI up-to-date Linux Amazon paling banyak yang tersedia pada saat rilis EMR Amazon. AMI diuji kompatibilitasnya dengan aplikasi big data dan fitur Amazon EMR yang disertakan dengan versi rilis tersebut.

Setiap Amazon EMR 5.35.0 dan yang lebih rendah, dan 6.5.0 dan versi rilis Amazon EMR yang lebih rendah “dikunci” ke versi AMI Amazon Linux masing-masing yang ditetapkan untuk mempertahankan kompatibilitas. Untuk alasan ini, kami menyarankan Anda menggunakan versi rilis Amazon EMR terbaru, kecuali Anda memerlukan versi yang lebih rendah untuk kompatibilitas dan tidak dapat bermigrasi. Jika Anda harus menggunakan versi rilis Amazon EMR yang lebih rendah untuk kompatibilitas, kami sarankan Anda menggunakan rilis terbaru dalam seri. Misalnya, jika Anda harus

menggunakan seri 5.12, gunakan 5.12.2 bukan 5.12.0 atau 5.12.1. Jika rilis baru tersedia dalam satu seri, pertimbangkan untuk memigrasikan aplikasi Anda ke rilis baru tersebut.

Untuk informasi selengkapnya tentang perilaku pembaruan otomatis yang diperkenalkan dengan Amazon EMR 5.36.0 dan yang lebih tinggi dan 6.6.0 dan yang lebih tinggi, lihat. Pembaruan Amazon Linux otomatis untuk rilis Amazon EMR

Perilaku boot default tidak termasuk pembaruan kernel

Saat instans Amazon EC2 dalam klaster didasarkan pada AMI Amazon Linux default yang digunakan Amazon EMR dalam melakukan booting untuk pertama kalinya, instans tersebut memeriksa repositori paket yang diaktifkan untuk Amazon Linux dan Amazon EMR untuk pembaruan perangkat lunak yang berlaku untuk versi AMI. Seperti instans Amazon EC2 lainnya, pembaruan keamanan penting dan penting dari repositori ini diinstal secara otomatis.

Namun, jika Anda menggunakan versi lama Amazon Linux AMI, pembaruan keamanan terbaru mungkin tidak diinstal secara otomatis. Ini karena repositori yang referensi cluster EMR Anda diperbaiki untuk setiap versi Amazon Linux AMI.

Perhatikan juga bahwa, dalam konfigurasi jaringan Anda, Anda harus mengizinkan jalan keluar HTTP dan HTTPS ke repositori Amazon Linux di Amazon S3. Jika tidak, pembaruan keamanan akan gagal. Untuk informasi selengkapnya, lihat Amazon Linux - Package repositori di Panduan Pengguna Amazon EC2 untuk Instans Linux. Secara default, paket perangkat lunak lain dan pembaruan kernel yang memerlukan reboot, termasuk NVIDIA dan CUDA, dikecualikan dari pengunduhan otomatis saat boot pertama.

> **Important**
>
> Cluster EMR yang menjalankan AL2023 menggunakan perilaku default Amazon Linux, dan Amazon Machine Images (AMI) Anda dikunci ke versi tertentu dari repositori Amazon Linux. Secara default, klaster Anda tidak akan secara otomatis menerima pembaruan keamanan perangkat lunak saat diluncurkan. Cluster Anda hanya berisi pembaruan yang tersedia di versi AL2023 AMI yang Anda pilih saat membuat klaster. Untuk informasi selengkapnya, lihat Memperbarui Amazon Linux 2023 di Panduan Pengguna Amazon Linux 2023.

> **Important**
>
> Cluster EMR yang menjalankan Amazon Linux atau Amazon Linux 2 Amazon Machine Images (AMI) menggunakan perilaku default Amazon Linux, dan tidak secara otomatis mengunduh dan menginstal pembaruan kernel penting dan kritis yang memerlukan reboot. Ini adalah perilaku yang sama dengan instans Amazon EC2 lainnya yang menjalankan AMI Amazon Linux default. Jika pembaruan perangkat lunak Amazon Linux baru yang memerlukan reboot (seperti pembaruan kernel, NVIDIA, dan CUDA) tersedia setelah rilis EMR Amazon tersedia, instance cluster EMR yang menjalankan AMI default tidak secara otomatis mengunduh dan menginstal pembaruan tersebut. Untuk mendapatkan pembaruan kernel, Anda dapat menyesuaikan Amazon EMR AMI menjadi gunakan Amazon Linux AMI terbaru.

Cluster diluncurkan dengan atau tanpa pembaruan

Perhatikan bahwa jika pembaruan perangkat lunak tidak dapat diinstal karena repositori paket tidak dapat dijangkau pada boot klaster pertama, instans klaster masih harus menyelesaikan peluncurannya. Untuk instans, repositori mungkin tidak dapat dijangkau karena S3 tidak tersedia untuk sementara, atau Anda mungkin memiliki aturan VPC atau firewall yang dikonfigurasi untuk memblokir akses.

Jangan lari **`sudo yum update`**

Saat Anda terhubung ke instans klaster menggunakan SSH, beberapa baris pertama output layar menyediakan tautan ke catatan rilis untuk AMI Amazon Linux yang digunakan instans, pemberitahuan versi AMI Amazon Linux terbaru, pemberitahuan nomor paket yang tersedia untuk pembaruan dari repositori yang diaktifkan, dan arahan untuk menjalankan `sudo yum update`.

> **Important**
>
> Kami sangat menyarankan agar Anda tidak menjalankan `sudo yum update` instance cluster, baik saat terhubung dengan SSH atau saat Anda menggunakan tindakan bootstrap. Hal ini mungkin menyebabkan ketidakcocokan karena semua paket diinstal tanpa pandang bulu.

Praktik terbaik pembaruan perangkat lunak

Praktik terbaik untuk mengelola pembaruan perangkat lunak

- Jika Anda menggunakan versi rilis Amazon EMR yang lebih rendah, pertimbangkan dan uji migrasi ke rilis terbaru sebelum memperbarui paket perangkat lunak.
- Jika Anda bermigrasi ke versi rilis yang lebih tinggi atau memutakhirkan paket perangkat lunak, uji implementasinya di lingkungan non-produksi terlebih dahulu. Opsi untuk mengkloning cluster dengan konsol EMR Amazon sangat membantu untuk ini.
- Evaluasi pembaruan perangkat lunak untuk aplikasi dan versi Amazon Linux AMI Anda secara individual. Hanya uji dan instal paket di lingkungan produksi yang Anda anggap mutlak diperlukan untuk postur keamanan, fungsionalitas aplikasi, atau kinerja Anda.
- Lihat Pusat Keamanan Amazon Linux untuk pembaruan.
- Hindari menginstal paket dengan menghubungkan ke instans klaster individu menggunakan SSH. Sebagai gantinya, gunakan tindakan bootstrap untuk menginstal dan memperbarui paket pada semua instans klaster jika diperlukan. Ini mengharuskan Anda mengakhiri klaster dan meluncurkannya kembali. Untuk informasi selengkapnya, lihat Buat tindakan bootstrap untuk menginstal perangkat lunak tambahan.

Menggunakan AMI kustom

Saat Anda menggunakan Amazon EMR 5.7.0 atau lebih tinggi, Anda dapat memilih untuk menentukan AMI Amazon Linux kustom alih-alih AMI Linux Amazon default untuk Amazon EMR. AMI kustom berguna jika Anda ingin melakukan hal berikut:

- Pra-instal aplikasi dan lakukan penyesuaian lain alih-alih menggunakan tindakan bootstrap. Hal ini dapat meningkatkan waktu mulai klaster dan menyederhanakan alur kerja startup. Untuk informasi lebih lanjut dan contoh, lihat Membuat AMI Amazon Linux kustom dari instans yang telah dikonfigurasi sebelumnya.
- Terapkan konfigurasi klaster dan simpul yang lebih canggih daripada yang diizinkan oleh tindakan bootstrap.
- Enkripsi volume perangkat root EBS (volume boot) instans EC2 di cluster Anda jika Anda menggunakan versi EMR Amazon yang lebih rendah dari 5.24.0. Seperti AMI default, ukuran volume root minimum untuk AMI khusus adalah 10 GiB untuk Amazon EMR rilis 6.9 dan lebih rendah, dan 15 GiB untuk Amazon EMR rilis 6.10 dan lebih tinggi. Untuk informasi selengkapnya, lihat Membuat AMI khusus dengan volume perangkat asal Amazon EBS terenkripsi.

> **Note**
>
> Dimulai dengan Amazon EMR versi 5.24.0, Anda dapat menggunakan opsi konfigurasi keamanan untuk mengenkripsi perangkat asal EBS dan volume penyimpanan ketika Anda menentukan AWS KMS sebagai penyedia kunci Anda. Untuk informasi selengkapnya, lihat Enkripsi disk lokal.

AMI kustom harus ada di AWS Wilayah yang sama tempat Anda membuat klaster. Ini juga harus cocok dengan arsitektur instans EC2. Misalnya, instance m5.xlarge memiliki arsitektur x86_64. Oleh karena itu, untuk menyediakan m5.xlarge menggunakan AMI kustom, AMI kustom Anda juga harus memiliki arsitektur x86_64. Demikian pula, untuk menyediakan instance m6g.xlarge, yang memiliki arsitektur arm64, AMI kustom Anda harus memiliki arsitektur arm64. Untuk informasi selengkapnya tentang mengidentifikasi AMI Linux untuk jenis instans Anda, lihat Menemukan AMI Linux di Panduan Pengguna Amazon EC2 untuk Instans Linux.

> **Important**
>
> Cluster EMR yang menjalankan Amazon Linux atau Amazon Linux 2 Amazon Machine Images (AMI) menggunakan perilaku default Amazon Linux, dan tidak secara otomatis mengunduh dan menginstal pembaruan kernel penting dan kritis yang memerlukan reboot. Ini adalah perilaku yang sama dengan instans Amazon EC2 lainnya yang menjalankan AMI Amazon Linux default. Jika pembaruan perangkat lunak Amazon Linux baru yang memerlukan reboot (seperti pembaruan kernel, NVIDIA, dan CUDA) tersedia setelah rilis EMR Amazon tersedia, instance cluster EMR yang menjalankan AMI default tidak secara otomatis mengunduh dan menginstal pembaruan tersebut. Untuk mendapatkan pembaruan kernel, Anda dapat menyesuaikan Amazon EMR AMI menjadi gunakan Amazon Linux AMI terbaru.

Membuat AMI Amazon Linux kustom dari instans yang telah dikonfigurasi sebelumnya

Langkah-langkah dasar untuk pra-instal perangkat lunak dan melakukan konfigurasi lain untuk membuat AMI Amazon Linux kustom untuk Amazon EMR adalah sebagai berikut:

- Luncurkan instans dari AMI Amazon Linux dasar.
- Connect ke instans untuk menginstal perangkat lunak dan melakukan penyesuaian lainnya.

- Buat citra baru (snapshot AMI) dari instans yang Anda konfigurasikan.

Setelah Anda membuat citra berdasarkan instans khusus, Anda dapat menyalin citra tersebut ke target terenkripsi seperti yang dijelaskan dalam Membuat AMI khusus dengan volume perangkat asal Amazon EBS terenkripsi.

Tutorial: Membuat AMI dari instans dengan perangkat lunak kustom yang telah diinstal

Untuk meluncurkan instans EC2 berdasarkan AMI Amazon Linux terbaru

1. Gunakan AWS CLI untuk menjalankan perintah berikut, yang membuat instans dari AMI yang sudah ada. Ganti *MyKeyName* dengan key pair yang Anda gunakan untuk menyambung ke instance dan *MyAmiId*dengan ID AMI Amazon Linux yang sesuai. Untuk ID AMI terbaru, lihat AMI Amazon Linux.

   > **Note**
   >
   > Karakter lanjutan baris Linux (\) disertakan agar mudah dibaca. Karakter ini bisa dihapus atau digunakan dalam perintah Linux. Untuk Windows, hapus atau ganti dengan tanda pangkat (^).

   ```
   aws ec2 run-instances --image-id MyAmiID \
   --count 1 --instance-type m5.xlarge \
   --key-name MyKeyName --region us-west-2
   ```

   Nilai output `InstanceId` digunakan sebagaimana *MyInstanceId* pada langkah berikutnya.

2. Jalankan perintah berikut:

   ```
   aws ec2 describe-instances --instance-ids MyInstanceId
   ```

   Nilai output `PublicDnsName` digunakan untuk menghubungkan ke instans pada langkah berikutnya.

Untuk terhubung ke instans dan menginstal perangkat lunak

1. Gunakan koneksi SSH yang memungkinkan Anda menjalankan perintah shell di instans Linux Anda. Untuk informasi lebih lanjut, lihat Menghubungkan ke instans Linux Anda menggunakan SSH di Panduan Pengguna Amazon EC2 untuk Instans Linux.
2. Lakukan penyesuaian yang diperlukan. Misalnya:

```
sudo yum install MySoftwarePackage
sudo pip install MySoftwarePackage
```

Untuk membuat snapshot dari citra kustom Anda

- Setelah Anda menyesuaikan instans, gunakan perintah `create-image` untuk membuat AMI dari instans.

```
aws ec2 create-image --no-dry-run --instance-id MyInstanceId --name MyEmrCustomAmi
```

Nilai output `imageID` digunakan saat Anda meluncurkan klasster atau membuat snapshot terenkripsi. Untuk informasi lebih lanjut, lihat Gunakan AMI kustom tunggal dalam kluster EMR dan Membuat AMI khusus dengan volume perangkat asal Amazon EBS terenkripsi.

Cara menggunakan AMI khusus di cluster EMR Amazon

Anda dapat menggunakan AMI khusus untuk menyediakan klaster EMR Amazon dengan dua cara:

- Gunakan satu AMI kustom untuk semua instans EC2 di cluster.
- Gunakan AMI kustom yang berbeda untuk berbagai jenis instans EC2 yang digunakan dalam cluster.

Anda hanya dapat menggunakan salah satu dari dua opsi saat menyediakan kluster EMR, dan Anda tidak dapat mengubahnya setelah cluster dimulai.

Pertimbangan untuk menggunakan AMI kustom tunggal versus beberapa di kluster EMR Amazon

Pertimbangan	AMI kustom tunggal	Beberapa AMI kustom
Gunakan prosesor x86 dan Graviton2 dengan AMI khusus di cluster yang sama	✗ Tidak didukung	✓ Didukung
Kustomisasi AMI bervariasi antar jenis instance	✗ Tidak didukung	✓ Didukung
Ubah AMI kustom saat menambahkan grup/armada instance tugas baru ke cluster yang sedang berjalan. Catatan: Anda tidak dapat mengubah AMI kustom grup/armada instans yang ada.	✗ Tidak didukung	✓ Didukung
Gunakan AWS Konsol untuk memulai klaster	✓ Didukung	✗ Tidak didukung
Gunakan AWS CloudFormation untuk memulai cluster	✓ Didukung	✓ Didukung

Gunakan AMI kustom tunggal dalam kluster EMR

Untuk menentukan ID AMI kustom saat Anda membuat klaster, gunakan salah satu dari berikut ini:

- AWS Management Console
- AWS CLI
- Amazon EMR SDK
- API EMR Amazon RunJobFlow
- AWS CloudFormation(lihat `CustomAmiID` properti di Cluster InstanceGroupConfig, Cluster InstanceTypeConfig, Resource InstanceGroupConfig, atau Resource InstanceFleetConfig - InstanceTypeConfig)

Amazon EMR console

Untuk menentukan AMI kustom tunggal dari konsol

1. Masuk keAWS Management Console, dan buka konsol EMR Amazon di https://console.aws.amazon.com/emr.
2. Di bawah EMR pada EC2 di panel navigasi kiri, pilih Clusters, lalu pilih Create cluster.
3. Di bawah Nama dan aplikasi, temukan Opsi sistem operasi. Pilih AMI Kustom, dan masukkan ID AMI Anda di bidang AMI Kustom.
4. Pilih opsi lain yang berlaku untuk cluster Anda.
5. Untuk meluncurkan klaster Anda, pilih Buat klaster.

AWS CLI

Untuk menentukan AMI kustom tunggal dengan AWS CLI

- Gunakan parameter `--custom-ami-id` untuk menentukan ID AMI saat Anda menjalankan perintah `aws emr create-cluster`.

  Contoh berikut menentukan cluster yang menggunakan AMI kustom tunggal dengan volume boot 20 GiB. Untuk informasi selengkapnya, lihat Menyesuaikan volume perangkat root Amazon EBS.

  > **Note**
  >
  > Karakter lanjutan baris Linux (\) disertakan agar mudah dibaca. Karakter ini bisa dihapus atau digunakan dalam perintah Linux. Untuk Windows, hapus atau ganti dengan tanda sisipan (^).

```
aws emr create-cluster --name "Cluster with My Custom AMI" \
--custom-ami-id MyAmiID --ebs-root-volume-size 20 \
--release-label emr-5.7.0 --use-default-roles \
--instance-count 2 --instance-type m5.xlarge
```

Gunakan beberapa AMI kustom di kluster EMR Amazon

Untuk membuat klaster menggunakan beberapa AMI kustom, gunakan salah satu dari berikut ini:

- AWSCLI versi 1.20.21 atau lebih tinggi
- SDK AWS
- Amazon EMR RunJobFlowdi Referensi API EMR Amazon
- AWS CloudFormation(lihat `CustomAmiID` properti di Cluster InstanceGroupConfig, Cluster InstanceTypeConfig, Resource InstanceGroupConfig, atau Resource InstanceFleetConfig - InstanceTypeConfig)

Konsol AWS Manajemen saat ini tidak mendukung pembuatan klaster menggunakan beberapa AMI kustom.

Example - Gunakan AWS CLI untuk membuat cluster grup instance menggunakan beberapa AMI kustom

Menggunakan AWS CLI versi 1.20.21 atau yang lebih tinggi, Anda dapat menetapkan satu AMI kustom ke seluruh cluster, atau Anda dapat menetapkan beberapa AMI kustom ke setiap node instance di cluster Anda.

Contoh berikut menunjukkan cluster grup instance seragam yang dibuat dengan dua tipe instance (m5.xlarge) yang digunakan di seluruh tipe node (primer, inti, tugas). Setiap node memiliki beberapa AMI khusus. Contoh ini mengilustrasikan beberapa fitur dari beberapa konfigurasi AMI kustom:

- Tidak ada AMI khusus yang ditetapkan di tingkat cluster. Ini untuk menghindari konflik antara beberapa AMI kustom dan satu AMI kustom, yang akan menyebabkan peluncuran cluster gagal.
- Cluster dapat memiliki beberapa AMI khusus di seluruh node tugas primer, inti, dan individu. Hal ini memungkinkan penyesuaian AMI individual, seperti aplikasi pra-instal, konfigurasi cluster canggih, dan volume perangkat root Amazon EBS terenkripsi.
- Node inti grup instance hanya dapat memiliki satu jenis instance dan AMI kustom yang sesuai. Demikian pula, node utama hanya dapat memiliki satu jenis instance dan AMI kustom yang sesuai.
- Cluster dapat memiliki beberapa node tugas.

```
aws emr create-cluster --instance-groups
InstanceGroupType=PRIMARY,InstanceType=m5.xlarge,InstanceCount=1,CustomAmiId=ami-123456
```

```
InstanceGroupType=CORE,InstanceType=m5.xlarge,InstanceCount=1,CustomAmiId=ami-234567
InstanceGroupType=TASK,InstanceType=m6g.xlarge,InstanceCount=1,CustomAmiId=ami-345678
InstanceGroupType=TASK,InstanceType=m5.xlarge,InstanceCount=1,CustomAmiId=ami-456789
```

Example - Gunakan AWS CLI versi 1.20.21 atau lebih tinggi untuk menambahkan node tugas ke cluster grup instance yang sedang berjalan dengan beberapa jenis instance dan beberapa AMI kustom

Menggunakan AWS CLI versi 1.20.21 atau yang lebih tinggi, Anda dapat menambahkan beberapa AMI kustom ke grup instans yang Anda tambahkan ke cluster yang sedang berjalan. `CustomAmiId`Argumen dapat digunakan dengan `add-instance-groups` perintah seperti yang ditunjukkan pada contoh berikut. Perhatikan bahwa beberapa ID AMI kustom yang sama (ami-123456) digunakan di lebih dari satu node.

```
aws emr create-cluster --instance-groups
InstanceGroupType=PRIMARY,InstanceType=m5.xlarge,InstanceCount=1,CustomAmiId=ami-123456
InstanceGroupType=CORE,InstanceType=m5.xlarge,InstanceCount=1,CustomAmiId=ami-123456
InstanceGroupType=TASK,InstanceType=m5.xlarge,InstanceCount=1,CustomAmiId=ami-234567

{
    "ClusterId": "j-123456",
    ...
}

aws emr add-instance-groups --cluster-id j-123456 --instance-groups
 InstanceGroupType=Task,InstanceType=m6g.xlarge,InstanceCount=1,CustomAmiId=ami-345678
```

Example - Gunakan AWS CLI versi 1.20.21 atau lebih tinggi untuk membuat cluster armada instance, beberapa AMI kustom, beberapa jenis instans, primer On-Demand, inti Sesuai Permintaan, beberapa inti dan node tugas

```
aws emr create-cluster --instance-fleets
InstanceFleetType=PRIMARY,TargetOnDemandCapacity=1,InstanceTypeConfigs=['{InstanceType=m5.xlarg
 CustomAmiId=ami-123456}']
InstanceFleetType=CORE,TargetOnDemandCapacity=1,InstanceTypeConfigs=['{InstanceType=m5.xlarge,C
{InstanceType=m6g.xlarge, CustomAmiId=ami-345678}']
InstanceFleetType=TASK,TargetSpotCapacity=1,InstanceTypeConfigs=['{InstanceType=m5.xlarge,Custo
{InstanceType=m6g.xlarge, CustomAmiId=ami-567890}']
```

Example - Gunakan AWS CLI versi 1.20.21 atau lebih tinggi untuk menambahkan node tugas ke cluster yang berjalan dengan beberapa jenis instance dan beberapa AMI kustom

```
aws emr create-cluster --instance-fleets
InstanceFleetType=PRIMARY,TargetOnDemandCapacity=1,InstanceTypeConfigs=['{InstanceType=m5.xlarg
 CustomAmiId=ami-123456}']
InstanceFleetType=CORE,TargetOnDemandCapacity=1,InstanceTypeConfigs=['{InstanceType=m5.xlarge,C
{InstanceType=m6g.xlarge, CustomAmiId=ami-345678}']

{
    "ClusterId": "j-123456",
    ...
}

aws emr add-instance-fleet --cluster-id j-123456 --instance-fleet
InstanceFleetType=TASK,TargetSpotCapacity=1,InstanceTypeConfigs=['{InstanceType=m5.xlarge,Custo
{InstanceType=m6g.xlarge, CustomAmiId=ami-345678}']
```

Mengelola pembaruan repositori paket AMI

Saat boot pertama, secara default, AMI Amazon Linux akan terhubung ke repositori paket untuk menginstal pembaruan keamanan sebelum layanan lain dimulai. Tergantung pada persyaratan Anda, Anda dapat memilih untuk menonaktifkan pembaruan ini saat Anda menentukan AMI kustom untuk Amazon EMR. Opsi untuk menonaktifkan fitur ini hanya tersedia saat Anda menggunakan AMI kustom. Secara default, pembaruan kernel Amazon Linux dan paket perangkat lunak lain yang mengharuskan boot ulang tidak diperbarui. Perhatikan bahwa konfigurasi jaringan Anda harus mengizinkan HTTP dan HTTPS keluar ke repositori Amazon Linux di Amazon S3, jika tidak, pembaruan keamanan tidak akan berhasil.

> **Warning**
>
> Kami sangat menyarankan Anda memilih untuk memperbarui semua paket yang diinstal saat boot ulang di mana Anda menentukan AMI kustom. Memilih untuk tidak memperbarui paket mengakibatkan risiko keamanan tambahan.

DenganAWS Management Console, Anda dapat memilih opsi untuk menonaktifkan pembaruan saat Anda memilih AMI Kustom.

DenganAWS CLI, Anda dapat menentukan `--repo-upgrade-on-boot NONE` bersama dengan `--custom-ami-id` saat menggunakan create-cluster perintah.

Dengan Amazon EMR API, Anda dapat menentukan `NONE` parameter. RepoUpgradeOnBoot

Membuat AMI khusus dengan volume perangkat asal Amazon EBS terenkripsi

Untuk mengenkripsi volume perangkat asal Amazon EBS dari AMI Amazon Linux untuk Amazon EMR, salin citra snapshot dari AMI yang tidak terenkripsi ke target terenkripsi. Untuk informasi lebih lanjut tentang cara membuat volume EBS, lihat Enkripsi Amazon EBS dalam Panduan Pengguna Amazon EC2 untuk Instans Linux. AMI sumber untuk snapshot dapat menjadi AMI Amazon Linux dasar, atau Anda dapat menyalin snapshot dari AMI yang berasal dari AMI Amazon Linux dasar yang Anda sesuaikan.

> **Note**
>
> Dimulai dengan Amazon EMR versi 5.24.0, Anda dapat menggunakan opsi konfigurasi keamanan untuk mengenkripsi perangkat asal EBS dan volume penyimpanan ketika Anda menentukan AWS KMS sebagai penyedia kunci Anda. Untuk informasi selengkapnya, lihat Enkripsi disk lokal.

Anda dapat menggunakan penyedia kunci eksternal atau kunci KMS AWS untuk mengenkripsi volume asal EBS. Peran layanan yang digunakan Amazon EMR (biasanya default`EMR_DefaultRole`) harus diizinkan untuk mengenkripsi dan mendekripsi volume, minimal, agar Amazon EMR membuat cluster dengan AMI. Saat menggunakan AWS KMS sebagai penyedia kunci, ini berarti bahwa tindakan berikut harus diizinkan:

- `kms:encrypt`
- `kms:decrypt`
- `kms:ReEncrypt*`
- `kms:CreateGrant`
- `kms:GenerateDataKeyWithoutPlaintext"`
- `kms:DescribeKey"`

Cara termudah untuk melakukan ini adalah dengan menambahkan peran sebagai pengguna kunci seperti yang dijelaskan dalam tutorial berikut. Contoh pernyataan kebijakan berikut diberikan jika Anda perlu menyesuaikan kebijakan peran.

```
{
"Version": "2012-10-17",
"Statement": [
   {
   "Sid": "EmrDiskEncryptionPolicy",
   "Effect": "Allow",
   "Action": [
      "kms:Encrypt",
      "kms:Decrypt",
      "kms:ReEncrypt*",
      "kms:CreateGrant",
      "kms:GenerateDataKeyWithoutPlaintext",
      "kms:DescribeKey"
      ],
   "Resource": [
      "*"
      ]
   }
]
}
```

Tutorial: Membuat AMI kustom dengan volume perangkat root terenkripsi menggunakan tombol KMS

Langkah pertama dalam contoh ini adalah menemukan ARN dari kunci KMS atau membuat yang baru. Untuk informasi selengkapnya tentang pembuatan kunci, lihat Membuat Kunci di AWS Key Management Service Panduan Developer. Prosedur berikut menunjukkan cara menambahkan peran layanan default, `EMR_DefaultRole`, sebagai pengguna kunci untuk kebijakan kunci. Tuliskan nilai ARN untuk kunci saat Anda membuat atau mengeditnya. Anda menggunakan ARN yang lebih tinggi, saat Anda membuat AMI.

Untuk menambahkan peran layanan untuk Amazon EC2 ke daftar pengguna kunci enkripsi dengan konsol

1. Masuk ke AWS Management Console dan buka konsol AWS Key Management Service (AWS KMS) di https://console.aws.amazon.com/kms.
2. Untuk mengubah Wilayah AWS, gunakan pemilih Wilayah di sudut kanan atas halaman.

3. Pilih alias tombol KMS yang akan digunakan.
4. Pada halaman detail kunci di bawah Pengguna Kunci, pilih Tambahkan.
5. Di kotak dialog Lampirkan, pilih peran layanan Amazon EMR. Nama peran default adalah `EMR_DefaultRole`.
6. Pilih Lampirkan.

Untuk membuat AMI terenkripsi dengan AWS CLI

- Gunakan perintah `aws ec2 copy-image` dari AWS CLI untuk membuat AMI dengan volume perangkat asal EBS terenkripsi dan kunci yang Anda ubah. Ganti `--kms-key-id` nilai yang ditentukan dengan ARN penuh dari kunci yang Anda buat atau modifikasi lebih rendah.

> **Note**
>
> Karakter lanjutan baris Linux (\) disertakan agar mudah dibaca Karakter ini bisa dihapus atau digunakan dalam perintah Linux. Untuk Windows, hapus atau ganti dengan tanda pangkat (^).

```
aws ec2 copy-image --source-image-id MyAmiId \
--source-region us-west-2 --name MyEncryptedEMRAmi \
--encrypted --kms-key-id arn:aws:kms:us-west-2:12345678910:key/xxxxxxxx-xxxx-xxxx-xxxx-xxxxxxxxxxxx
```

Output dari perintah memberikan ID AMI yang Anda buat, yang dapat Anda tentukan saat membuat klaster. Untuk informasi selengkapnya, lihat Gunakan AMI kustom tunggal dalam kluster EMR. Anda juga dapat memilih untuk menyesuaikan AMI ini dengan menginstal perangkat lunak dan melakukan konfigurasi lainnya. Untuk informasi selengkapnya, lihat Membuat AMI Amazon Linux kustom dari instans yang telah dikonfigurasi sebelumnya.

Praktik terbaik dan pertimbangan

Saat Anda membuat AMI kustom untuk Amazon EMR, pertimbangkan hal-hal berikut:

- Amazon EMR 5.30.0 dan lebih tinggi, dan seri Amazon EMR 6.x didasarkan pada Amazon Linux 2. Untuk versi EMR Amazon ini, Anda perlu menggunakan gambar berdasarkan Amazon Linux 2 untuk AMI khusus. Untuk menemukan AMI kustom dasar, lihat Menemukan AMI Linux.
- Untuk Amazon EMR versi lebih rendah dari 5.30.0 dan 6.x, Amazon Linux 2 AMI tidak didukung.
- Anda harus menggunakan AMI Amazon Linux 64-bit. AMI 32-bit tidak didukung.
- AMI Amazon Linux dengan beberapa volume Amazon EBS tidak didukung.
- Dasarkan penyesuaian Anda pada AMI Amazon Linux terbaru yang didukung EBS. Untuk daftar AMI Amazon Linux dan ID AMI yang sesuai, lihat AMI Amazon Linux.
- Jangan menyalin snapshot instans Amazon EMR yang ada untuk membuat AMI kustom. Hal ini dapat menyebabkan kesalahan.
- Hanya jenis virtualisasi HVM dan instans yang kompatibel dengan Amazon EMR yang didukung. Pastikan untuk memilih gambar HVM dan jenis instans yang kompatibel dengan Amazon EMR saat Anda menjalani proses penyesuaian AMI. Untuk contoh yang kompatibel dan jenis virtualisasi, lihat Tipe instans yang didukung.
- Peran layanan Anda harus memiliki izin peluncuran di AMI, jadi AMI harus bersifat publik, atau Anda harus menjadi pemilik AMI atau dibagikan kepada Anda oleh pemiliknya.
- Membuat pengguna di AMI dengan nama yang sama dengan aplikasi menyebabkan kesalahan (misalnya,`hadoop`, `hdfs`, `yarn`, atau `spark`).
- Isi dari `/tmp`, `/var`, dan `/emr` (jika mereka ada di AMI) dipindahkan ke masing-masing `/mnt/tmp`, `/mnt/var`, dan `/mnt/emr` selama startup. File disimpan, tetapi jika terdapat banyak data, startup mungkin memerlukan waktu lebih lama dari yang diperkirakan.
- Jika Anda menggunakan AMI Amazon Linux khusus berdasarkan AMI Amazon Linux dengan tanggal pembuatan 2018-08-11, server Oozie gagal memulai. Jika Anda menggunakan Oozie, buat AMI kustom berdasarkan ID AMI Amazon Linux dengan tanggal pembuatan yang berbeda. Anda dapat menggunakan AWS CLI perintah berikut untuk mengembalikan daftar ID Gambar untuk semua AMI Linux Amazon HVM dengan versi 2018.03, bersama dengan tanggal rilis, sehingga Anda dapat memilih AMI Amazon Linux yang sesuai sebagai basis Anda. Ganti MyRegion dengan pengenal Wilayah Anda, seperti us-west-2.

```
aws ec2 --region MyRegion describe-images --owner amazon --query 'Images[?
Name!=`null`]|[?starts_with(Name, `amzn-ami-hvm-2018.03`) == `true`].
[CreationDate,ImageId,Name]' --output text | sort -rk1
```

- Dalam kasus di mana Anda menggunakan VPC dengan nama domain non-standar dan AmazonProvided DNS, Anda tidak boleh menggunakan `rotate` opsi dalam konfigurasi DNS Sistem Operasi.

Untuk informasi selengkapnya, lihat Membuat AMI Linux yang didukung Amazon EBS di Panduan Pengguna Amazon EC2 untuk Instans Linux.

Mengubah rilis Amazon Linux saat Anda membuat klaster EMR

Saat Anda meluncurkan cluster menggunakan Amazon EMR 6.6.0 atau lebih tinggi, secara otomatis menggunakan rilis Amazon Linux 2 terbaru yang telah divalidasi untuk Amazon EMR AMI default. Anda dapat menentukan rilis Amazon Linux yang berbeda untuk klaster Anda dengan konsol Amazon EMR atau. AWS CLI

Amazon EMR console

Untuk mengubah rilis Amazon Linux saat Anda membuat cluster dari konsol

1. Masuk keAWS Management Console, dan buka konsol EMR Amazon di https://console.aws.amazon.com/emr.
2. Di bawah EMR pada EC2 di panel navigasi kiri, pilih Clusters, lalu pilih Create cluster.
3. Untuk versi EMR, pilih emr-6.6.0 atau lebih tinggi.
4. Di bawah Opsi sistem operasi, pilih versi Amazon Linux, dan pilih kotak centang Terapkan pembaruan Amazon Linux terbaru secara otomatis.
5. Pilih opsi lain yang berlaku untuk cluster Anda.
6. Untuk meluncurkan klaster Anda, pilih Buat klaster.

AWS CLI

Untuk mengubah rilis Amazon Linux saat Anda membuat cluster dengan AWS CLI

- Gunakan `--os-release-label` parameter untuk menentukan Rilis Amazon Linux saat Anda menjalankan perintah `aws emr` create-cluster.

```
aws emr create-cluster --name "Cluster with Different Amazon Linux Release" \
--os-release-label 2.0.20210312.1 \
--release-label emr-6.6.0 --use-default-roles \
```

```
--instance-count 2 --instance-type m5.xlarge
```

Menyesuaikan volume perangkat root Amazon EBS

Default volume root EBS

Dengan Amazon EMR 4.x dan yang lebih tinggi, Anda dapat menentukan ukuran volume root saat membuat cluster. Dengan Amazon EMR rilis 6.15.0 dan yang lebih tinggi, Anda juga dapat menentukan IOPS volume root dan throughput. Atribut hanya berlaku untuk volume perangkat root Amazon EBS, dan berlaku untuk semua instance di cluster. Atribut tidak berlaku untuk volume penyimpanan, yang Anda tentukan secara terpisah untuk setiap jenis instans saat membuat klaster.

- Ukuran volume root default adalah 15 GiB di Amazon EMR 6.10.0 dan lebih tinggi. Rilis sebelumnya memiliki ukuran volume root default 10 GiB. Anda dapat menyesuaikan ini hingga 100 GiB.
- Volume root default IOPS adalah 3000. Anda dapat menyesuaikan ini hingga 16000.
- Output volume root default adalah 125 MiB/s. Anda dapat menyesuaikan ini hingga 1000 MiB/s.

> **Note**
>
> Ukuran volume root dan IOPS tidak dapat memiliki rasio lebih tinggi dari 1 volume hingga 500 IOPS (1:500), sedangkan volume root IOPS dan throughput tidak dapat memiliki rasio yang lebih tinggi dari 1 IOPS hingga 0,25 throughput (1:0,25).

Untuk informasi selengkapnya tentang Amazon EBS, lihat Volume perangkat asal Amazon EC2.

Jenis volume perangkat root dengan AMI default

Saat Anda menggunakan AMI default, jenis volume perangkat root ditentukan oleh rilis EMR Amazon yang Anda gunakan.

- Dengan Amazon EMR merilis 6.15.0 dan lebih tinggi, Amazon EMR memasang General Purpose SSD (gp3) sebagai jenis volume perangkat root.
- Dengan rilis Amazon EMR lebih rendah dari 6.15.0, Amazon EMR memasang General Purpose SSD (gp2) sebagai jenis volume perangkat root.

Jenis volume perangkat root dengan AMI khusus

AMI khusus mungkin memiliki jenis volume perangkat root yang berbeda. Amazon EMR selalu menggunakan tipe volume AMI kustom Anda.

- Dengan Amazon EMR rilis 6.15.0 dan yang lebih tinggi, Anda dapat mengonfigurasi ukuran volume root, IOPS, dan throughput untuk AMI kustom Anda, asalkan atribut ini berlaku untuk jenis volume AMI kustom.
- Dengan rilis Amazon EMR yang lebih rendah dari 6.15.0, Anda hanya dapat mengonfigurasi ukuran volume root untuk AMI kustom Anda.

Jika Anda tidak mengonfigurasi ukuran volume root, IOPS, atau throughput saat membuat klaster, Amazon EMR menggunakan nilai dari AMI kustom jika berlaku. Jika Anda memutuskan untuk mengonfigurasi nilai-nilai ini saat membuat klaster, Amazon EMR menggunakan nilai yang Anda tentukan selama nilainya kompatibel dan didukung oleh volume root AMI kustom. Untuk informasi selengkapnya, lihat Menggunakan AMI kustom.

Harga ukuran volume perangkat root

Biaya volume perangkat root EBS dinilai per jam, berdasarkan biaya EBS bulanan untuk jenis volume tersebut di Wilayah tempat cluster berjalan. Hal yang sama berlaku untuk volume penyimpanan. Biaya dalam GB, tetapi Anda menentukan ukuran volume root di GiB, jadi Anda mungkin ingin mempertimbangkan ini dalam perkiraan Anda (1 GB adalah 0,931323 GiB).

General Purpose SSD gp2 dan gp3 ditagih secara berbeda. Untuk memperkirakan biaya yang terkait dengan volume perangkat root EBS di cluster Anda, gunakan rumus berikut:

Tujuan Umum SSD gp2

Biaya untuk gp2 hanya mencakup ukuran volume EBS dalam GB.

```
($EBS size in GB/month) * 0.931323 / 30 / 24 * EMR_EBSRootVolumesizeInGiB *
 InstanceCount
```

Misalnya, ambil cluster yang memiliki node primer, node inti, dan menggunakan basis Amazon Linux AMI, dengan volume perangkat root 10 GiB default. Jika biaya EBS di Wilayah adalah USD $0,10/GB/bulan, itu berarti sekitar $0,00129 per instans per jam, dan $0,00258 per jam untuk cluster ($0,10/GB/bulan dibagi 30 hari, dibagi 24 jam, dikalikan dengan 10 GB, dikalikan dengan 2 instance cluster).

Tujuan Umum SSD gp3

Biaya untuk gp3 termasuk ukuran volume EBS dalam GB, IOPS di atas 3000 (3000 IOPS gratis), dan throughput di atas 125 MB/s (125 MB/s gratis).

```
($EBS size in GB/month) * 0.931323 / 30 / 24 * EMR_EBSRootVolumesizeInGiB *
 InstanceCount
+
($EBS IOPS/Month)/30/24* (EMR_EBSRootVolumeIops - 3000) * InstanceCount
+
($EBS throughput/Month)/30/24* (EMR_EBSRootVolumeThroughputInMb/s - 125) *
 InstanceCount
```

Misalnya, ambil cluster yang memiliki node primer, node inti, dan menggunakan basis Amazon Linux AMI, dengan ukuran volume perangkat root 15 GiB default, 4000 IOPS, dan 140 throughput. Jika biaya EBS di Wilayah adalah USD $0,10/GB/bulan, $0,005/IOP yang disediakan/bulan di atas 3000, dan $0,040/MB/s/bulan yang disediakan di atas 125. Itu berarti sekitar $0,009293 per instance per jam, dan $0,018586 per jam untuk cluster.

Menentukan pengaturan volume perangkat root kustom

> **Note**
>
> Ukuran volume root dan IOPS tidak dapat memiliki rasio lebih tinggi dari 1 volume hingga 500 IOPS (1:500), sedangkan volume root IOPS dan throughput tidak dapat memiliki rasio yang lebih tinggi dari 1 IOPS hingga 0,25 throughput (1:0,25).

Console

Untuk menentukan atribut volume perangkat root Amazon EBS dari konsol EMR Amazon

1. Masuk keAWS Management Console, dan buka konsol EMR Amazon di https://console.aws.amazon.com/emr.
2. Di bawah EMR pada EC2 di panel navigasi kiri, pilih Clusters, lalu pilih Create cluster.
3. Pilih Amazon EMR rilis 6.15.0 atau lebih tinggi.
4. Di bawah konfigurasi Cluster, navigasikan ke bagian volume root EBS dan masukkan nilai untuk atribut apa pun yang ingin Anda konfigurasikan.

5. Pilih opsi lain yang berlaku untuk cluster Anda.
6. Untuk meluncurkan klaster Anda, pilih Buat klaster.

CLI

Untuk menentukan atribut volume perangkat root Amazon EBS dengan AWS CLI

- Gunakan`--ebs-root-volume-size`,`--ebs-root-volume-iops`, dan `--ebs-root-volume-throughput` parameter perintah create-cluster, seperti yang ditunjukkan pada contoh berikut.

> **Note**
>
> Karakter lanjutan baris Linux (\) disertakan agar mudah dibaca Karakter ini bisa dihapus atau digunakan dalam perintah Linux. Untuk Windows, hapus atau ganti dengan tanda pangkat (^).

```
aws emr create-cluster --release-label emr-6.15.0\
--ebs-root-volume-size 20 \
--ebs-root-volume-iops 3000\
--ebs-root-volume-throughput 135\
--instance-groups InstanceGroupType=MASTER,\
InstanceCount=1,InstanceType=m5.xlarge
 InstanceGroupType=CORE,InstanceCount=2,InstanceType=m5.xlarge
```

Konfigurasi perangkat lunak klaster

Saat Anda memilih rilis perangkat lunak, Amazon EMR menggunakan Amazon Machine Image (AMI) dengan Amazon Linux untuk menginstal perangkat lunak yang Anda pilih saat meluncurkan klaster, seperti Hadoop, Spark, dan Hive. Amazon EMR menyediakan rilis baru secara berkala, menambahkan fitur baru, aplikasi baru, dan pembaruan umum. Kami menyarankan Anda menggunakan rilis terbaru untuk meluncurkan klaster bila memungkinkan. Rilis terbaru adalah opsi default jika Anda meluncurkan klaster dari konsol.

Untuk informasi selengkapnya tentang rilis Amazon EMR dan versi perangkat lunak yang tersedia dengan setiap rilis, buka Panduan Rilis Amazon EMR. Untuk informasi selengkapnya tentang

cara mengedit konfigurasi default aplikasi dan perangkat lunak yang diinstal di klaster Anda, buka Mengonfigurasi aplikasi di Panduan Rilis Amazon EMR. Beberapa versi komponen ekosistem Hadoop dan Spark sumber terbuka yang disertakan dalam rilis Amazon EMR memiliki patch dan peningkatan, yang mana didokumentasikan dalam Panduan Rilis Amazon EMR.

Selain perangkat lunak dan aplikasi standar yang tersedia untuk diinstal di klaster, Anda dapat menggunakan tindakan bootstrap untuk menginstal perangkat lunak kustom. Tindakan bootstrap adalah skrip yang berjalan pada instans saat klaster Anda diluncurkan, dan yang berjalan pada simpul baru yang ditambahkan ke klaster Anda saat dibuat. Tindakan bootstrap juga berguna untuk menjalankan perintah AWS CLI pada setiap simpul untuk menyalin objek dari Amazon S3 ke setiap simpul di klaster Anda.

> **Note**
>
> Tindakan bootstrap digunakan secara berbeda-beda di Amazon EMR rilis 4.x dan yang lebih baru. Untuk informasi lebih lanjut tentang perbedaan ini dari AMI Amazon EMR versi 2.x dan 3.x, buka Perbedaan yang diperkenalkan di 4.x di Panduan Rilis Amazon EMR.

Buat tindakan bootstrap untuk menginstal perangkat lunak tambahan

Anda dapat menggunakan Tindakan bootstrap untuk menginstal perangkat lunak tambahan atau menyesuaikan konfigurasi instans klaster. Tindakan bootstrap adalah skrip yang berjalan di klaster setelah Amazon EMR meluncurkan instans menggunakan Amazon Machine Image (AMI) Amazon Linux. Tindakan bootstrap dijalankan sebelum Amazon EMR menginstal aplikasi yang Anda tentukan saat membuat klaster dan sebelum simpul klaster mulai memproses data. Jika Anda menambahkan simpul ke klaster yang sedang berjalan, tindakan bootstrap juga berjalan pada simpul tersebut dengan cara yang sama. Anda dapat membuat tindakan bootstrap kustom dan menentukannya saat membuat klaster.

Sebagian besar tindakan bootstrap yang telah ditentukan sebelumnya untuk AMI Amazon EMR versi 2.x dan 3.x tidak didukung di Amazon EMR rilis 4.x. Misalnya, `configure-Hadoop` dan `configure-daemons` tidak didukung di Amazon EMR rilis 4.x. Sebaliknya, Amazon EMR release 4.x secara native menyediakan fungsionalitas ini. Untuk informasi lebih lanjut tentang cara memigrasikan tindakan bootstrap dari Amazon EMR AMI versi 2.x dan 3.x ke Amazon EMR rilis 4.x, buka Menyesuaikan cluster dan konfigurasi aplikasi dengan versi AMI sebelumnya dari Amazon EMR di Panduan Rilis Amazon EMR Amazon.

Dasar-dasar tindakan bootstrap

Tindakan bootstrap dijalankan sebagai pengguna Hadoop secara default. Anda dapat menjalankan tindakan bootstrap dengan hak akses root menggunakan `sudo`.

Semua antarmuka manajemen Amazon EMR mendukung tindakan bootstrap. Anda dapat menentukan hingga 16 tindakan bootstrap per klaster dengan menyediakan beberapa parameter `bootstrap-actions` dari konsol, AWS CLI, atau API.

Dari konsol Amazon EMR, Secara opsional, Anda dapat menentukan tindakan bootstrap saat membuat klaster.

Saat menggunakan CLI, Anda dapat meneruskan referensi skrip tindakan bootstrap ke Amazon EMR dengan menambahkan parameter `--bootstrap-actions` saat Anda membuat klaster menggunakan perintah `create-cluster`.

```
--bootstrap-actions Path="s3://mybucket/filename",Args=[arg1,arg2]
```

Jika tindakan bootstrap mengembalikan kode kesalahan bukan nol, Amazon EMR memperlakukannya sebagai kegagalan dan mengakhiri instans. Jika terlalu banyak instans yang gagal dalam tindakan bootstrapnya, Amazon EMR akan mengakhiri klaster. Jika hanya beberapa instans yang gagal, Amazon EMR akan mencoba mengalokasikan ulang instans yang gagal dan melanjutkannya. Gunakan kode kesalahan `lastStateChangeReason` klaster untuk mengidentifikasi kegagalan yang disebabkan oleh tindakan bootstrap.

Jalankan tindakan bootstrap secara kondisional

Untuk hanya menjalankan tindakan bootstrap pada node master, Anda dapat menggunakan tindakan bootstrap khusus dengan beberapa logika untuk menentukan apakah node tersebut master.

```
#!/bin/bash
if grep isMaster /mnt/var/lib/info/instance.json | grep false;
then
    echo "This is not master node, do nothing,exiting"
    exit 0
fi
echo "This is master, continuing to execute script"
# continue with code logic for master node below
```

Output berikut akan mencetak dari node inti.

```
This is not master node, do nothing, exiting
```

Output berikut akan mencetak dari master node.

```
This is master, continuing to execute script
```

Untuk menggunakan logika ini, unggah tindakan bootstrap Anda, termasuk kode di atas, ke bucket Amazon S3 Anda. PadaAWS CLI, tambahkan `--bootstrap-actions` parameter ke panggilan `aws emr create-cluster` API dan tentukan lokasi skrip bootstrap Anda sebagai nilai`Path`.

Tindakan penghentian

Skrip tindakan bootstrap dapat membuat satu atau lebih tindakan penghentian dengan menulis skrip ke direktori `/mnt/var/lib/instance-controller/public/shutdown-actions/`. Ketika sebuah cluster diakhiri, semua skrip di direktori ini dijalankan secara paralel. Setiap skrip harus dijalankan dan diselesaikan dalam waktu 60 detik.

Skrip tindakan penghentian tidak dijamin berjalan jika simpul diakhiri karena kesalahan.

> **Note**
>
> Saat menggunakan Amazon EMR versi 4.0 dan yang lebih baru, Anda harus membuat direktori `/mnt/var/lib/instance-controller/public/shutdown-actions/` secara manual di simpul utama. Ini tidak ada secara default; namun, setelah dibuat, skrip di direktori ini tetap berjalan sebelum dihentikan. Untuk informasi lebih lanjut tentang menghubungkan ke Simpul utama untuk membuat direktori, lihat Connect ke node utama menggunakan SSH.

Gunakan tindakan bootstrap kustom

Anda dapat membuat skrip kustom untuk melakukan tindakan bootstrap yang disesuaikan. Antarmuka Amazon EMR mana pun dapat mereferensikan tindakan bootstrap kustom.

> **Note**
>
> Untuk kinerja terbaik, kami menyarankan Anda menyimpan tindakan bootstrap kustom, skrip, dan file lain yang ingin Anda gunakan dengan Amazon EMR di bucket Amazon S3 yang sama dengan cluster Anda. Wilayah AWS

Daftar Isi

Tambahkan tindakan bootstrap kustom

> **Note**
>
> Kami telah mendesain ulang konsol EMR Amazon agar lebih mudah digunakan. Lihat Apa yang baru dengan konsol? untuk mempelajari tentang perbedaan antara pengalaman konsol lama dan baru.

New console

Untuk membuat cluster dengan aksi bootstrap dengan konsol baru

1. Masuk keAWS Management Console, dan buka konsol EMR Amazon di https://console.aws.amazon.com/emr.
2. Di bawah EMR pada EC2 di panel navigasi kiri, pilih Clusters, lalu pilih Create cluster.
3. Di bawah Tindakan Bootstrap, pilih Tambahkan untuk menentukan nama, lokasi skrip, dan argumen opsional untuk tindakan Anda. Pilih Tambahkan tindakan bootstrap.
4. Secara opsional, tambahkan lebih banyak tindakan bootstrap.
5. Pilih opsi lain yang berlaku untuk cluster Anda.
6. Untuk meluncurkan klaster Anda, pilih Buat klaster.

Old console

Untuk membuat cluster dengan aksi bootstrap khusus dengan konsol lama

1. Arahkan ke konsol EMR Amazon baru dan pilih Beralih ke konsol lama dari navigasi samping. Untuk informasi selengkapnya tentang apa yang diharapkan saat beralih ke konsol lama, lihat Menggunakan konsol lama.
2. Pilih Buat klaster.
3. Klik Pergi ke opsi lanjutan.

4. Di Buat Klaster - Opsi Lanjutan, Langkah 1 dan 2 pilih opsi yang diinginkan dan lanjutkan ke Langkah 3: Pengaturan Klaster Umum.
5. Di bawah Tindakan Bootstrap pilih Konfigurasi dan tambahkan untuk menentukan Nama, lokasi JAR, dan argumen untuk tindakan bootstrap Anda. Pilih Tambahkan.
6. Secara opsional tambahkan lebih banyak tindakan bootstrap sesuai keinginan.
7. Lanjutkan untuk membuat klaster. Tindakan bootstrap Anda akan dilakukan setelah klaster telah disediakan dan diinisialisasi.

   Selama node utama cluster berjalan, Anda dapat terhubung ke node utama dan melihat file log yang dibuat oleh skrip tindakan bootstrap di `/mnt/var/log/bootstrap-actions/1` direktori.

CLI

Untuk membuat cluster dengan aksi bootstrap khusus dengan AWS CLI

Saat menggunakan AWS CLI untuk menyertakan tindakan bootstrap, tentukan `Path` dan `Args` sebagai daftar yang dipisahkan koma. Contoh berikut tidak menggunakan daftar argumen.

- Untuk meluncurkan cluster dengan aksi bootstrap kustom, ketik perintah berikut, ganti *myKey dengan nama key* pair EC2 Anda. Sertakan `--bootstrap-actions` sebagai parameter dan tentukan lokasi skrip bootstrap Anda sebagai nilai`Path`.

  - Pengguna Linux, UNIX, dan Mac OS X:

    ```
    aws emr create-cluster --name "Test cluster" --release-label emr-4.0.0 \
    --use-default-roles --ec2-attributes KeyName=myKey \
    --applications Name=Hive Name=Pig \
    --instance-count 3 --instance-type m5.xlarge \
    --bootstrap-actions Path="s3://elasticmapreduce/bootstrap-actions/download.sh"
    ```

  - Pengguna Windows:

    ```
    aws emr create-cluster --name "Test cluster" --release-label emr-4.2.0 --use-
    default-roles --ec2-attributes KeyName=myKey --applications Name=Hive Name=Pig
     --instance-count 3 --instance-type m5.xlarge --bootstrap-actions Path="s3://
    elasticmapreduce/bootstrap-actions/download.sh"
    ```

Saat Anda menentukan jumlah instance tanpa menggunakan `--instance-groups` parameter, satu node primer diluncurkan, dan instance yang tersisa diluncurkan sebagai node inti. Semua simpul akan menggunakan tipe instans yang ditentukan dalam perintah.

> **Note**
>
> Jika Anda belum pernah membuat peran layanan Amazon EMR default dan profil instans EC2, ketik `aws emr create-default-roles` untuk membuatnya sebelum mengetik `create-cluster` subperintah.

Untuk informasi selengkapnya tentang menggunakan perintah Amazon EMR dalam AWS CLI, lihat https://docs.aws.amazon.com/cli/latest/reference/emr.

Gunakan tindakan bootstrap kustom untuk menyalin objek dari Amazon S3 ke setiap simpul

Anda dapat menggunakan tindakan bootstrap untuk menyalin objek dari Amazon S3 ke setiap simpul dalam klaster sebelum aplikasi Anda diinstal. AWS CLI diinstal pada setiap simpul dari sebuah klaster, sehingga tindakan bootstrap Anda dapat memanggil perintah AWS CLI.

Contoh berikut menunjukkan skrip tindakan bootstrap sederhana yang menyalin file, `myfile.jar`, dari Amazon S3 ke folder lokal, `/mnt1/myfolder`, pada setiap simpul klaster. Skrip disimpan ke Amazon S3 dengan nama file `copymyfile.sh` yang berisi konten berikut.

```
#!/bin/bash
aws s3 cp s3://mybucket/myfilefolder/myfile.jar /mnt1/myfolder
```

Saat Anda meluncurkan klaster, Anda menentukan skrip. Contoh AWS CLI berikut menunjukkan ini:

```
aws emr create-cluster --name "Test cluster" --release-label emr-7.0.0 \
--use-default-roles --ec2-attributes KeyName=myKey \
--applications Name=Hive Name=Pig \
--instance-count 3 --instance-type m5.xlarge \
--bootstrap-actions Path="s3://mybucket/myscriptfolder/copymyfile.sh"
```

Konfigurasi perangkat keras dan jaringan klaster

Pertimbangan penting saat membuat klaster EMR Amazon adalah cara mengonfigurasi instans Amazon EC2 dan opsi jaringan. Bab ini mencakup opsi-opsi berikut, dan kemudian mengikat semuanya bersama-sama dengan praktik terbaik dan panduan.

- Tipe node — Instans Amazon EC2 dalam kluster EMR diatur ke dalam tipe node. Ada tiga: node primer, node inti, dan node tugas. Setiap jenis simpul melakukan serangkaian peran yang ditentukan oleh aplikasi terdistribusi yang Anda instal di klaster. Selama pekerjaan Hadoop MapReduce atau Spark, misalnya, komponen pada inti dan node tugas memproses data, mentransfer output ke Amazon S3 atau HDFS, dan memberikan metadata status kembali ke node utama. Dengan cluster simpul tunggal, semua komponen berjalan pada simpul utama. Untuk informasi selengkapnya, lihat Memahami jenis node: node primer, inti, dan tugas.
- Instans EC2 — Saat membuat klaster, Anda membuat pilihan tentang instans Amazon EC2 yang akan dijalankan oleh setiap jenis node. Jenis instans EC2 menentukan profil pemrosesan dan penyimpanan simpul. Pilihan instans Amazon EC2 untuk node Anda penting karena menentukan profil kinerja masing-masing tipe node di cluster Anda. Untuk informasi selengkapnya, lihat Konfigurasikan instans Amazon EC2.
- Jaringan - Anda dapat meluncurkan cluster EMR Amazon Anda ke dalam VPC menggunakan subnet publik, subnet pribadi, atau subnet bersama. Konfigurasi jaringan Anda menentukan bagaimana pelanggan dan layanan dapat terhubung ke klaster untuk melakukan pekerjaan, bagaimana klaster terhubung ke penyimpanan data dan sumber daya AWS lainnya, dan opsi yang Anda miliki untuk mengontrol lalu lintas koneksi tersebut. Untuk informasi selengkapnya, lihat Mengkonfigurasi jaringan.
- Pengelompokan instans – Kumpulan instans EC2 yang menghosting setiap jenis simpul disebut Armada instans ataugrup instans seragam. Konfigurasi pengelompokan instans adalah pilihan yang Anda buat saat membuat klaster. Pilihan ini menentukan bagaimana Anda dapat menambahkan simpul ke klaster Anda saat sedang dijalankan. Konfigurasi ini berlaku untuk semua jenis simpul. Hal ini tidak dapat diubah nanti. Untuk informasi selengkapnya, lihat Membuat sebuah klaster dengan armada instan atau grup instans seragam.

> **Note**
>
> Konfigurasi armada instance hanya tersedia di Amazon EMR rilis 4.8.0 dan yang lebih baru, tidak termasuk 5.0.0 dan 5.0.3.

Memahami jenis node: node primer, inti, dan tugas

Gunakan bagian ini untuk memahami bagaimana Amazon EMR menggunakan setiap jenis simpul ini dan sebagai dasar untuk perencanaan kapasitas klaster.

Node utama

Node primer mengelola cluster dan biasanya menjalankan komponen utama dari aplikasi terdistribusi. Misalnya, node utama menjalankan ResourceManager layanan YARN untuk mengelola sumber daya untuk aplikasi. Ini juga menjalankan NameNode layanan HDFS, melacak status pekerjaan yang dikirimkan ke cluster, dan memantau kesehatan grup instans.

Untuk memantau kemajuan cluster dan berinteraksi langsung dengan aplikasi, Anda dapat terhubung ke node utama melalui SSH sebagai pengguna Hadoop. Untuk informasi selengkapnya, lihat Connect ke node utama menggunakan SSH. Menghubungkan ke node utama memungkinkan Anda untuk mengakses direktori dan file, seperti file log Hadoop, secara langsung. Untuk informasi selengkapnya, lihat Melihat berkas log. Anda juga dapat melihat antarmuka pengguna yang diterbitkan aplikasi sebagai situs web yang berjalan di simpul utama. Untuk informasi selengkapnya, lihat Melihat antarmuka web yang di-host pada klaster Amazon EMR.

> **Note**
>
> Dengan Amazon EMR 5.23.0 dan yang lebih baru, Anda dapat meluncurkan cluster dengan tiga node utama untuk mendukung ketersediaan aplikasi yang tinggi seperti YARN Resource Manager, HDFS, Spark, Hive NameNode, dan Ganglia. Node primer tidak lagi menjadi titik kegagalan tunggal potensial dengan fitur ini. Jika salah satu node primer gagal, Amazon EMR secara otomatis gagal ke node primer siaga dan mengganti node primer yang gagal dengan yang baru dengan konfigurasi dan tindakan bootstrap yang sama. Untuk informasi selengkapnya, lihat Merencanakan dan Mengkonfigurasi Node Utama.

Simpul inti

Node inti dikelola oleh simpul utama. Simpul inti menjalankan daemon Simpul Data untuk mengoordinasikan penyimpanan data sebagai bagian dari Sistem File Terdistribusi Hadoop (HDFS). Mereka juga menjalankan daemon Task Tracker dan melakukan tugas komputasi paralel lainnya pada data yang diperlukan oleh aplikasi yang diinstal. Misalnya, node inti menjalankan NodeManager daemon YARN, MapReduce tugas Hadoop, dan pelaksana Spark.

Hanya ada satu grup instans inti atau armada instans per cluster, tetapi mungkin ada beberapa node yang berjalan di beberapa instans Amazon EC2 di grup instans atau armada instans. Dengan grup instans, Anda dapat menambahkan dan menghapus instans Amazon EC2 saat klaster sedang berjalan. Anda juga dapat menyiapkan penskalaan otomatis untuk menambahkan instans berdasarkan nilai metrik. Untuk informasi selengkapnya tentang menambahkan dan menghapus instans Amazon EC2 dengan konfigurasi grup instans, lihat. Gunakan penskalaan cluster

Dengan armada instans, Anda dapat secara efektif menambah dan menghapus instans dengan memodifikasi kapasitas target armada instans untuk Sesuai Permintaan dan Spot sebagaimana mestinya. Untuk informasi selengkapnya tentang kapasitas target, lihat Opsi armada instans.

> **Warning**
>
> Menghapus daemon HDFS dari simpul inti yang sedang berjalan atau mengakhiri simpul inti mengakibatkan risiko kehilangan data. Berhati-hatilah saat mengonfigurasi simpul inti untuk menggunakan Instans Spot. Untuk informasi selengkapnya, lihat Kapan Anda harus menggunakan Instans Spot?.

Simpul tugas

Anda dapat menggunakan node tugas untuk menambahkan daya untuk melakukan tugas komputasi paralel pada data, seperti tugas Hadoop MapReduce dan pelaksana Spark. Simpul tugas tidak menjalankan daemon Simpul Dat, juga tidak menyimpan data dalam HDFS. Seperti halnya node inti, Anda dapat menambahkan node tugas ke klaster dengan menambahkan instans Amazon EC2 ke grup instans seragam yang ada atau dengan memodifikasi kapasitas target untuk armada instance tugas.

Dengan konfigurasi grup instans seragam, Anda dapat memiliki hingga total 48 grup instans tugas. Kemampuan untuk menambahkan grup instans dengan cara ini memungkinkan Anda untuk menggabungkan jenis instans Amazon EC2 dan opsi harga, seperti Instans Sesuai Permintaan dan Instans Spot. Ini memberi Anda fleksibilitas untuk menanggapi persyaratan beban kerja dengan cara yang hemat biaya.

Dengan konfigurasi armada instans, kemampuan untuk memadukan jenis instans dan opsi pembelian sudah ada di dalamnya, sehingga hanya ada satu armada instans tugas.

Karena Instans Spot sering digunakan untuk menjalankan simpul tugas, Amazon EMR memiliki fungsionalitas default untuk menjadwalkan tugas YARN sehingga tugas yang sedang berjalan tidak

mengalami kegagalan saat simpul tugas yang berjalan pada Instans Spot diakhiri. Amazon EMR melakukan ini dengan mengizinkan proses utama aplikasi berjalan hanya pada simpul inti. Proses utama aplikasi mengontrol tugas yang sedang berjalan dan harus tetap hidup selama masa tugas.

Amazon EMR merilis 5.19.0 dan yang lebih baru menggunakan fitur label node YARN bawaan untuk mencapai ini. (Versi sebelumnya menggunakan patch kode). Properti dalam klasifikasi konfigurasi `yarn-site` dan `capacity-scheduler` dikonfigurasi secara default sehingga YARN capacity-scheduler dan fair-scheduler memanfaatkan label simpul. Amazon EMR secara otomatis melabeli simpul inti dengan label `CORE`, dan menetapkan properti sehingga utama aplikasi dijadwalkan hanya pada simpul dengan label INTI. Mengubah properti terkait secara manual dalam klasifikasi konfigurasi yarn-site dan capacity-scheduler, atau secara langsung dalam file XML terkait, dapat merusak fitur ini atau mengubah fungsionalitas ini.

Dimulai dengan Amazon EMR seri rilis 6.x, fitur label simpul YARN dinonaktifkan secara default. Proses utama aplikasi dapat berjalan pada node inti dan tugas secara default. Anda dapat mengaktifkan fitur label simpul YARN dengan mengkonfigurasi properti berikut:

- `yarn.node-labels.enabled: true`
- `yarn.node-labels.am.default-node-label-expression: 'CORE'`

Untuk informasi tentang properti tertentu, lihat Pengaturan Amazon EMR untuk mencegah kegagalan tugas karena pengakhiran Instans Spot simpul tugas.

Konfigurasikan instans Amazon EC2

Instans EC2 hadir dalam konfigurasi berbeda yang dikenal sebagai tipe instans. Tipe instans memiliki CPU, input/output, dan kapasitas penyimpanan yang berbeda. Selain jenis instans, Anda dapat memilih opsi pembelian yang berbeda untuk instans Amazon EC2. Anda dapat menentukan berbagai tipe instans dan opsi pembelian dalam grup instans seragam atau armada instans. Untuk informasi selengkapnya, lihat Membuat sebuah klaster dengan armada instan atau grup instans seragam. Untuk panduan tentang memilih tipe instans dan opsi pembelian untuk aplikasi Anda, lihat Praktik terbaik untuk konfigurasi klaster.

> **Important**
>
> Saat Anda memilih tipe instans menggunakan AWS Management Console, jumlah vCPU yang ditampilkan untuk setiap Tipe instans adalah jumlah vcore YARN untuk tipe instans

tersebut, bukan jumlah vCPU EC2 untuk tipe instans tersebut. Untuk informasi selengkapnya tentang jumlah vCPU untuk setiap tipe instans, lihat Tipe Instans Amazon EC2.

Topik

- Tipe instans yang didukung
- Mengkonfigurasi jaringan
- Membuat sebuah klaster dengan armada instan atau grup instans seragam

Tipe instans yang didukung

Bagian ini menjelaskan jenis instans yang didukung Amazon EMR, yang diatur oleh. Wilayah AWS Untuk mempelajari lebih lanjut tentang jenis instans, lihat instans Amazon EC2 dan matriks tipe instans Amazon Linux AMI.

Tidak semua tipe instans tersedia di semua Wilayah, ketersediaan instans bergantung pada ketersediaan dan permintaan di Wilayah dan Availability Zone yang ditentukan. Availability Zone instance ditentukan oleh subnet yang Anda gunakan untuk meluncurkan cluster Anda.

Pertimbangan-pertimbangan

Pertimbangkan hal berikut ketika Anda memilih jenis instans untuk klaster EMR Amazon Anda.

Important

Saat Anda memilih tipe instans menggunakan AWS Management Console, jumlah vCPU yang ditampilkan untuk setiap Tipe instans adalah jumlah vcore YARN untuk tipe instans tersebut, bukan jumlah vCPU EC2 untuk tipe instans tersebut. Untuk informasi selengkapnya tentang jumlah vCPU untuk setiap tipe instans, lihat Tipe Instans Amazon EC2.

- Jika Anda membuat klaster menggunakan jenis instans yang tidak tersedia di Wilayah dan Zona Ketersediaan yang ditentukan, klaster Anda mungkin gagal menyediakan atau mungkin macet dalam penyediaan. Untuk informasi tentang ketersediaan instans, lihat halaman harga Amazon EMR atau lihat Jenis instans yang didukung oleh Wilayah AWS tabel di halaman ini.
- Muali dari Amazon EMR versi rilis 5.13.0, semua instans menggunakan virtualisasi HVM dan penyimpanan yang didukung EBS untuk volume asal. Jika versi rilis Amazon EMR lebih awal dari

5.13.0 digunakan, beberapa instans generasi sebelumnya menggunakan virtualisasi PVM. Untuk informasi selengkapnya, lihat Jenis virtualisasi Linux AMI.

- Beberapa jenis instans mendukung jaringan yang ditingkatkan. Untuk informasi lebih lanjut, lihat Jaringan yang Ditingkatkan pada Linux.
- Driver NVIDIA dan CUDA diinstal pada tipe instans GPU secara default.
- Untuk Amazon EMR 7.0.0 release (`emr-7.0.0`), kami telah menemukan kemungkinan masalah korupsi data saat Anda menjalankan lowongan di kluster EMR yang menggunakan tipe instans EC2 tertentu. Ini karena masalah mendasar di JDK 17. Hingga masalah ini teratasi, jenis instance berikut untuk sementara tidak didukung di Amazon EMR 7.0.0. Kami akan mengaktifkan dukungan untuk jenis instans ini dalam pembaruan yang akan datang.

  `c6i`, `c6id`, `c6in`, `c7a`, `c7i`, `i4i`, `m6i`, `m6id`, `m6idn`, `m6in`, `m7a`, `m7i`, `m7i-flex`, `r6i`, `r6id`, `r6idn`, `r6in`, `r7a`, `r7i`, `r7iz`, `x2idn`, `x2iedn`

Jenis instans yang didukung oleh Wilayah AWS

Tabel berikut mencantumkan jenis instans Amazon EC2 yang didukung Amazon EMR, yang diatur oleh. Wilayah AWS Tabel juga mencantumkan rilis EMR Amazon paling awal dalam seri 5.x, 6.x, dan 7.x yang mendukung setiap jenis instans.

US East (N. Virginia) - us-east-1

Kelas Instans	Tipe instans	Versi EMR Amazon yang didukung minimum (5.x, 6.x, 7.x)
Tujuan Umum	m5.xlarge	emr-5.13.0, emr-6.0.0, emr-7.0.0
	m5.2xlarge	emr-5.13.0, emr-6.0.0, emr-7.0.0
	m5.4xlarge	emr-5.13.0, emr-6.0.0, emr-7.0.0
	m5.8xlarge	emr-5.13.0, emr-6.0.0, emr-7.0.0

Kelas Instans	Tipe instans	Versi EMR Amazon yang didukung minimum (5.x, 6.x, 7.x)
	m5.12xlarge	emr-5.13.0, emr-6.0.0, emr-7.0.0
	m5.16xlarge	emr-5.13.0, emr-6.0.0, emr-7.0.0
	m5.24xlarge	emr-5.13.0, emr-6.0.0, emr-7.0.0
	m5a.xlarge	emr-5.20.0, emr-6.0.0, emr-7.0.0
	m5a.2xlarge	emr-5.20.0, emr-6.0.0, emr-7.0.0
	m5a.4xlarge	emr-5.20.0, emr-6.0.0, emr-7.0.0
	m5a.8xlarge	emr-5.20.0, emr-6.0.0, emr-7.0.0
	m5a.12xlarge	emr-5.20.0, emr-6.0.0, emr-7.0.0
	m5a.16xlarge	emr-5.20.0, emr-6.0.0, emr-7.0.0
	m5a.24xlarge	emr-5.20.0, emr-6.0.0, emr-7.0.0
	m5ad.xlarge	emr-5.20.0, emr-6.0.0, emr-7.0.0
	m5ad.2xlarge	emr-5.20.0, emr-6.0.0, emr-7.0.0

Kelas Instans	Tipe instans	Versi EMR Amazon yang didukung minimum (5.x, 6.x, 7.x)
	m5ad.4xlarge	emr-5.20.0, emr-6.0.0, emr-7.0.0
	m5ad.8xlarge	emr-5.33.0, emr-6.3.0, emr-7.0.0
	m5ad.12xlarge	emr-5.20.0, emr-6.0.0, emr-7.0.0
	m5ad.16xlarge	emr-5.33.0, emr-6.3.0, emr-7.0.0
	m5ad.24xlarge	emr-5.20.0, emr-6.0.0, emr-7.0.0
	m5d.xlarge	emr-5.13.0, emr-6.0.0, emr-7.0.0
	m5d.2xlarge	emr-5.13.0, emr-6.0.0, emr-7.0.0
	m5d.4xlarge	emr-5.13.0, emr-6.0.0, emr-7.0.0
	m5d.8xlarge	emr-5.13.0, emr-6.0.0, emr-7.0.0
	m5d.12xlarge	emr-5.13.0, emr-6.0.0, emr-7.0.0
	m5d.16xlarge	emr-5.13.0, emr-6.0.0, emr-7.0.0
	m5d.24xlarge	emr-5.13.0, emr-6.0.0, emr-7.0.0

Kelas Instans	Tipe instans	Versi EMR Amazon yang didukung minimum (5.x, 6.x, 7.x)
	m5dn.xlarge	emr-5.31.0, emr-6.1.0, emr-7.0.0
	m5dn.2xlarge	emr-5.31.0, emr-6.1.0, emr-7.0.0
	m5dn.4xlarge	emr-5.31.0, emr-6.1.0, emr-7.0.0
	m5dn.8xlarge	emr-5.31.0, emr-6.1.0, emr-7.0.0
	m5dn.12xlarge	emr-5.31.0, emr-6.1.0, emr-7.0.0
	m5dn.16xlarge	emr-5.31.0, emr-6.1.0, emr-7.0.0
	m5dn.24xlarge	emr-5.31.0, emr-6.1.0, emr-7.0.0
	m5n.xlarge	emr-5.31.0, emr-6.1.0, emr-7.0.0
	m5n.2xlarge	emr-5.31.0, emr-6.1.0, emr-7.0.0
	m5n.4xlarge	emr-5.31.0, emr-6.1.0, emr-7.0.0
	m5n.8xlarge	emr-5.31.0, emr-6.1.0, emr-7.0.0
	m5n.12xlarge	emr-5.31.0, emr-6.1.0, emr-7.0.0

Kelas Instans	Tipe instans	Versi EMR Amazon yang didukung minimum (5.x, 6.x, 7.x)
	m5n.16xlarge	emr-5.31.0, emr-6.1.0, emr-7.0.0
	m5n.24xlarge	emr-5.31.0, emr-6.1.0, emr-7.0.0
	m5zn.xlarge	emr-5.33.0, emr-6.3.0, emr-7.0.0
	m5zn.2xlarge	emr-5.33.0, emr-6.3.0, emr-7.0.0
	m5zn.3xlarge	emr-5.33.0, emr-6.3.0, emr-7.0.0
	m5zn.6xlarge	emr-5.33.0, emr-6.3.0, emr-7.0.0
	m5zn.12xlarge	emr-5.33.0, emr-6.3.0, emr-7.0.0
	m6a.xlarge	emr-5.36.1, emr-6.8.0, emr-7.0.0
	m6a.2xlarge	emr-5.36.1, emr-6.8.0, emr-7.0.0
	m6a.4xlarge	emr-5.36.1, emr-6.8.0, emr-7.0.0
	m6a.8xlarge	emr-5.36.1, emr-6.8.0, emr-7.0.0
	m6a.12xlarge	emr-5.36.1, emr-6.8.0, emr-7.0.0

Kelas Instans	Tipe instans	Versi EMR Amazon yang didukung minimum (5.x, 6.x, 7.x)
	m6a.16xlarge	emr-5.36.1, emr-6.8.0, emr-7.0.0
	m6a.24xlarge	emr-5.36.1, emr-6.8.0, emr-7.0.0
	m6a.32xlarge	emr-5.36.1, emr-6.8.0, emr-7.0.0
	m6a.48xlarge	emr-5.36.1, emr-6.8.0, emr-7.0.0
	m6g.xlarge	emr-5.30.0, emr-6.1.0, emr-7.0.0
	m6g.2xlarge	emr-5.30.0, emr-6.1.0, emr-7.0.0
	m6g.4xlarge	emr-5.30.0, emr-6.1.0, emr-7.0.0
	m6g.8xlarge	emr-5.30.0, emr-6.1.0, emr-7.0.0
	m6g.12xlarge	emr-5.30.0, emr-6.1.0, emr-7.0.0
	m6g.16xlarge	emr-5.30.0, emr-6.1.0, emr-7.0.0
	m6gd.xlarge	emr-5.33.0, emr-6.3.0, emr-7.0.0
	m6gd.2xlarge	emr-5.33.0, emr-6.3.0, emr-7.0.0

Kelas Instans	Tipe instans	Versi EMR Amazon yang didukung minimum (5.x, 6.x, 7.x)
	m6gd.4xlarge	emr-5.33.0, emr-6.3.0, emr-7.0.0
	m6gd.8xlarge	emr-5.33.0, emr-6.3.0, emr-7.0.0
	m6gd.12xlarge	emr-5.33.0, emr-6.3.0, emr-7.0.0
	m6gd.16xlarge	emr-5.33.0, emr-6.3.0, emr-7.0.0
	m6i.xlarge	emr-5.35.0, emr-6.6.0
	m6i.2xlarge	emr-5.35.0, emr-6.6.0
	m6i.4xlarge	emr-5.35.0, emr-6.6.0
	m6i.8xlarge	emr-5.35.0, emr-6.6.0
	m6i.12xlarge	emr-5.35.0, emr-6.6.0
	m6i.16xlarge	emr-5.35.0, emr-6.6.0
	m6i.24xlarge	emr-5.35.0, emr-6.6.0
	m6i.32xlarge	emr-5.35.0, emr-6.6.0
	m6id.xlarge	emr-5.36.1, emr-6.8.0
	m6id.2xlarge	emr-5.36.1, emr-6.8.0
	m6id.4xlarge	emr-5.36.1, emr-6.8.0
	m6id.8xlarge	emr-5.36.1, emr-6.8.0
	m6id.12xlarge	emr-5.36.1, emr-6.8.0

Kelas Instans	Tipe instans	Versi EMR Amazon yang didukung minimum (5.x, 6.x, 7.x)
	m6id.16xlarge	emr-5.36.1, emr-6.8.0
	m6id.24xlarge	emr-5.36.1, emr-6.8.0
	m6id.32xlarge	emr-5.36.1, emr-6.8.0
	m6idn.xlarge	emr-5.36.1, emr-6.10.0
	m6idn.2xlarge	emr-5.36.1, emr-6.10.0
	m6idn.4xlarge	emr-5.36.1, emr-6.10.0
	m6idn.8xlarge	emr-5.36.1, emr-6.10.0
	m6idn.12xlarge	emr-5.36.1, emr-6.10.0
	m6idn.16xlarge	emr-5.36.1, emr-6.10.0
	m6idn.24xlarge	emr-5.36.1, emr-6.10.0
	m6idn.32xlarge	emr-5.36.1, emr-6.10.0
	m6in.xlarge	emr-5.36.1, emr-6.10.0
	m6in.2xlarge	emr-5.36.1, emr-6.10.0
	m6in.4xlarge	emr-5.36.1, emr-6.10.0
	m6in.8xlarge	emr-5.36.1, emr-6.10.0
	m6in.12xlarge	emr-5.36.1, emr-6.10.0
	m6in.16xlarge	emr-5.36.1, emr-6.10.0
	m6in.24xlarge	emr-5.36.1, emr-6.10.0
	m6in.32xlarge	emr-5.36.1, emr-6.10.0

Kelas Instans	Tipe instans	Versi EMR Amazon yang didukung minimum (5.x, 6.x, 7.x)
	m7a.xlarge	emr-5.36.1, emr-6.10.0
	m7a.2xlarge	emr-5.36.1, emr-6.10.0
	m7a.4xlarge	emr-5.36.1, emr-6.10.0
	m7a.8xlarge	emr-5.36.1, emr-6.10.0
	m7a.12xlarge	emr-5.36.1, emr-6.10.0
	m7a.16xlarge	emr-5.36.1, emr-6.10.0
	m7a.24xlarge	emr-5.36.1, emr-6.10.0
	m7a.32xlarge	emr-5.36.1, emr-6.10.0
	m7a.48xlarge	emr-5.36.1, emr-6.10.0
	m7g.xlarge	emr-5.36.1, emr-6.10.0, emr-7.0.0
	m7g.2xlarge	emr-5.36.1, emr-6.10.0, emr-7.0.0
	m7g.4xlarge	emr-5.36.1, emr-6.10.0, emr-7.0.0
	m7g.8xlarge	emr-5.36.1, emr-6.10.0, emr-7.0.0
	m7g.12xlarge	emr-5.36.1, emr-6.10.0, emr-7.0.0
	m7g.16xlarge	emr-5.36.1, emr-6.10.0, emr-7.0.0

Kelas Instans	Tipe instans	Versi EMR Amazon yang didukung minimum (5.x, 6.x, 7.x)
	m7gd.xlarge	emr-5.36.1, emr-6.10.0, emr-7.0.0
	m7gd.2xlarge	emr-5.36.1, emr-6.10.0, emr-7.0.0
	m7gd.4xlarge	emr-5.36.1, emr-6.10.0, emr-7.0.0
	m7gd.8xlarge	emr-5.36.1, emr-6.10.0, emr-7.0.0
	m7gd.12xlarge	emr-5.36.1, emr-6.10.0, emr-7.0.0
	m7gd.16xlarge	emr-5.36.1, emr-6.10.0, emr-7.0.0
	m7i.xlarge	emr-5.36.1, emr-6.10.0
	m7i.2xlarge	emr-5.36.1, emr-6.10.0
	m7i.4xlarge	emr-5.36.1, emr-6.10.0
	m7i.8xlarge	emr-5.36.1, emr-6.10.0
	m7i.12xlarge	emr-5.36.1, emr-6.10.0
	m7i.16xlarge	emr-5.36.1, emr-6.10.0
	m7i.24xlarge	emr-5.36.1, emr-6.10.0
	m7i.48xlarge	emr-5.36.1, emr-6.10.0
	m7i-flex.xlarge	emr-5.36.1, emr-6.10.0
	m7i-flex.2xlarge	emr-5.36.1, emr-6.10.0

Kelas Instans	Tipe instans	Versi EMR Amazon yang didukung minimum (5.x, 6.x, 7.x)
	m7i-flex.4xlarge	emr-5.36.1, emr-6.10.0
	m7i-flex.8xlarge	emr-5.36.1, emr-6.10.0
Komputasi Dioptimalkan	c5.xlarge	emr-5.13.0, emr-6.0.0, emr-7.0.0
	c5.2xlarge	emr-5.13.0, emr-6.0.0, emr-7.0.0
	c5.4xlarge	emr-5.13.0, emr-6.0.0, emr-7.0.0
	c5.9xlarge	emr-5.13.0, emr-6.0.0, emr-7.0.0
	c5.12xlarge	emr-5.13.0, emr-6.0.0, emr-7.0.0
	c5.18xlarge	emr-5.13.0, emr-6.0.0, emr-7.0.0
	c5.24xlarge	emr-5.13.0, emr-6.0.0, emr-7.0.0
	c5a.xlarge	emr-5.31.0, emr-6.1.0, emr-7.0.0
	c5a.2xlarge	emr-5.31.0, emr-6.1.0, emr-7.0.0
	c5a.4xlarge	emr-5.31.0, emr-6.1.0, emr-7.0.0
	c5a.8xlarge	emr-5.31.0, emr-6.1.0, emr-7.0.0

Kelas Instans	Tipe instans	Versi EMR Amazon yang didukung minimum (5.x, 6.x, 7.x)
	c5a.12xlarge	emr-5.31.0, emr-6.1.0, emr-7.0.0
	c5a.16xlarge	emr-5.31.0, emr-6.1.0, emr-7.0.0
	c5a.24xlarge	emr-5.31.0, emr-6.1.0, emr-7.0.0
	c5ad.xlarge	emr-5.33.0, emr-6.3.0, emr-7.0.0
	c5ad.2xlarge	emr-5.33.0, emr-6.3.0, emr-7.0.0
	c5ad.4xlarge	emr-5.33.0, emr-6.3.0, emr-7.0.0
	c5ad.8xlarge	emr-5.33.0, emr-6.3.0, emr-7.0.0
	c5ad.12xlarge	emr-5.33.0, emr-6.3.0, emr-7.0.0
	c5ad.16xlarge	emr-5.33.0, emr-6.3.0, emr-7.0.0
	c5ad.24xlarge	emr-5.33.0, emr-6.3.0, emr-7.0.0
	c5d.xlarge	emr-5.13.0, emr-6.0.0, emr-7.0.0
	c5d.2xlarge	emr-5.13.0, emr-6.0.0, emr-7.0.0

Kelas Instans	Tipe instans	Versi EMR Amazon yang didukung minimum (5.x, 6.x, 7.x)
	c5d.4xlarge	emr-5.13.0, emr-6.0.0, emr-7.0.0
	c5d.9xlarge	emr-5.13.0, emr-6.0.0, emr-7.0.0
	c5d.12xlarge	emr-5.13.0, emr-6.0.0, emr-7.0.0
	c5d.18xlarge	emr-5.13.0, emr-6.0.0, emr-7.0.0
	c5d.24xlarge	emr-5.13.0, emr-6.0.0, emr-7.0.0
	c5n.xlarge	emr-5.20.0, emr-6.0.0, emr-7.0.0
	c5n.2xlarge	emr-5.20.0, emr-6.0.0, emr-7.0.0
	c5n.4xlarge	emr-5.20.0, emr-6.0.0, emr-7.0.0
	c5n.9xlarge	emr-5.20.0, emr-6.0.0, emr-7.0.0
	c5n.18xlarge	emr-5.20.0, emr-6.0.0, emr-7.0.0
	c6a.xlarge	emr-5.36.1, emr-6.8.0, emr-7.0.0
	c6a.2xlarge	emr-5.36.1, emr-6.8.0, emr-7.0.0

Kelas Instans	Tipe instans	Versi EMR Amazon yang didukung minimum (5.x, 6.x, 7.x)
	c6a.4xlarge	emr-5.36.1, emr-6.8.0, emr-7.0.0
	c6a.8xlarge	emr-5.36.1, emr-6.8.0, emr-7.0.0
	c6a.12xlarge	emr-5.36.1, emr-6.8.0, emr-7.0.0
	c6a.16xlarge	emr-5.36.1, emr-6.8.0, emr-7.0.0
	c6a.24xlarge	emr-5.36.1, emr-6.8.0, emr-7.0.0
	c6a.32xlarge	emr-5.36.1, emr-6.8.0, emr-7.0.0
	c6a.48xlarge	emr-5.36.1, emr-6.8.0, emr-7.0.0
	c6g.xlarge	emr-5.31.0, emr-6.1.0, emr-7.0.0
	c6g.2xlarge	emr-5.31.0, emr-6.1.0, emr-7.0.0
	c6g.4xlarge	emr-5.31.0, emr-6.1.0, emr-7.0.0
	c6g.8xlarge	emr-5.31.0, emr-6.1.0, emr-7.0.0
	c6g.12xlarge	emr-5.31.0, emr-6.1.0, emr-7.0.0

Kelas Instans	Tipe instans	Versi EMR Amazon yang didukung minimum (5.x, 6.x, 7.x)
	c6g.16xlarge	emr-5.31.0, emr-6.1.0, emr-7.0.0
	c6gd.xlarge	emr-5.33.0, emr-6.3.0, emr-7.0.0
	c6gd.2xlarge	emr-5.33.0, emr-6.3.0, emr-7.0.0
	c6gd.4xlarge	emr-5.33.0, emr-6.3.0, emr-7.0.0
	c6gd.8xlarge	emr-5.33.0, emr-6.3.0, emr-7.0.0
	c6gd.12xlarge	emr-5.33.0, emr-6.3.0, emr-7.0.0
	c6gd.16xlarge	emr-5.33.0, emr-6.3.0, emr-7.0.0
	c6gn.xlarge	emr-5.33.0, emr-6.3.0, emr-7.0.0
	c6gn.2xlarge	emr-5.33.0, emr-6.3.0, emr-7.0.0
	c6gn.4xlarge	emr-5.33.0, emr-6.3.0, emr-7.0.0
	c6gn.8xlarge	emr-5.33.0, emr-6.3.0, emr-7.0.0
	c6gn.12xlarge	emr-5.33.0, emr-6.3.0, emr-7.0.0

Kelas Instans	Tipe instans	Versi EMR Amazon yang didukung minimum (5.x, 6.x, 7.x)
	c6gn.16xlarge	emr-5.33.0, emr-6.3.0, emr-7.0.0
	c6i.xlarge	emr-5.35.0, emr-6.6.0
	c6i.2xlarge	emr-5.35.0, emr-6.6.0
	c6i.4xlarge	emr-5.35.0, emr-6.6.0
	c6i.8xlarge	emr-5.35.0, emr-6.6.0
	c6i.12xlarge	emr-5.35.0, emr-6.6.0
	c6i.16xlarge	emr-5.35.0, emr-6.6.0
	c6i.24xlarge	emr-5.35.0, emr-6.6.0
	c6i.32xlarge	emr-5.35.0, emr-6.6.0
	c6id.xlarge	emr-5.36.1, emr-6.8.0
	c6id.2xlarge	emr-5.36.1, emr-6.8.0
	c6id.4xlarge	emr-5.36.1, emr-6.8.0
	c6id.8xlarge	emr-5.36.1, emr-6.8.0
	c6id.12xlarge	emr-5.36.1, emr-6.8.0
	c6id.16xlarge	emr-5.36.1, emr-6.8.0
	c6id.24xlarge	emr-5.36.1, emr-6.8.0
	c6id.32xlarge	emr-5.36.1, emr-6.8.0
	c6in.xlarge	emr-5.36.1, emr-6.10.0
	c6in.2xlarge	emr-5.36.1, emr-6.10.0

Kelas Instans	Tipe instans	Versi EMR Amazon yang didukung minimum (5.x, 6.x, 7.x)
	c6in.4xlarge	emr-5.36.1, emr-6.10.0
	c6in.8xlarge	emr-5.36.1, emr-6.10.0
	c6in.12xlarge	emr-5.36.1, emr-6.10.0
	c6in.16xlarge	emr-5.36.1, emr-6.10.0
	c6in.24xlarge	emr-5.36.1, emr-6.10.0
	c6in.32xlarge	emr-5.36.1, emr-6.10.0
	c7a.xlarge	emr-5.36.1, emr-6.10.0
	c7a.2xlarge	emr-5.36.1, emr-6.10.0
	c7a.4xlarge	emr-5.36.1, emr-6.10.0
	c7a.8xlarge	emr-5.36.1, emr-6.10.0
	c7a.12xlarge	emr-5.36.1, emr-6.10.0
	c7a.16xlarge	emr-5.36.1, emr-6.10.0
	c7a.24xlarge	emr-5.36.1, emr-6.10.0
	c7a.32xlarge	emr-5.36.1, emr-6.10.0
	c7a.48xlarge	emr-5.36.1, emr-6.10.0
	c7g.xlarge	emr-5.36.1, emr-6.7.0, emr-7.0.0
	c7g.2xlarge	emr-5.36.1, emr-6.7.0, emr-7.0.0

Kelas Instans	Tipe instans	Versi EMR Amazon yang didukung minimum (5.x, 6.x, 7.x)
	c7g.4xlarge	emr-5.36.1, emr-6.7.0, emr-7.0.0
	c7g.8xlarge	emr-5.36.1, emr-6.7.0, emr-7.0.0
	c7g.12xlarge	emr-5.36.1, emr-6.7.0, emr-7.0.0
	c7g.16xlarge	emr-5.36.1, emr-6.7.0, emr-7.0.0
	c7gd.xlarge	emr-5.36.1, emr-6.10.0, emr-7.0.0
	c7gd.2xlarge	emr-5.36.1, emr-6.10.0, emr-7.0.0
	c7gd.4xlarge	emr-5.36.1, emr-6.10.0, emr-7.0.0
	c7gd.8xlarge	emr-5.36.1, emr-6.10.0, emr-7.0.0
	c7gd.12xlarge	emr-5.36.1, emr-6.10.0, emr-7.0.0
	c7gd.16xlarge	emr-5.36.1, emr-6.10.0, emr-7.0.0
	c7gn.xlarge	emr-5.36.1, emr-6.10.0, emr-7.0.0
	c7gn.2xlarge	emr-5.36.1, emr-6.10.0, emr-7.0.0

Kelas Instans	Tipe instans	Versi EMR Amazon yang didukung minimum (5.x, 6.x, 7.x)
	c7gn.4xlarge	emr-5.36.1, emr-6.10.0, emr-7.0.0
	c7gn.8xlarge	emr-5.36.1, emr-6.10.0, emr-7.0.0
	c7gn.12xlarge	emr-5.36.1, emr-6.10.0, emr-7.0.0
	c7gn.16xlarge	emr-5.36.1, emr-6.10.0, emr-7.0.0
	c7i.xlarge	emr-5.36.1, emr-6.10.0
	c7i.2xlarge	emr-5.36.1, emr-6.10.0
	c7i.4xlarge	emr-5.36.1, emr-6.10.0
	c7i.8xlarge	emr-5.36.1, emr-6.10.0
	c7i.12xlarge	emr-5.36.1, emr-6.10.0
	c7i.16xlarge	emr-5.36.1, emr-6.10.0
	c7i.24xlarge	emr-5.36.1, emr-6.10.0
	c7i.48xlarge	emr-5.36.1, emr-6.10.0
Komputasi yang Dipercepat	g3.4xlarge	emr-5.18.0, emr-6.0.0, emr-7.0.0
	g3.8xlarge	emr-5.18.0, emr-6.0.0, emr-7.0.0
	g3.16xlarge	emr-5.18.0, emr-6.0.0, emr-7.0.0

Kelas Instans	Tipe instans	Versi EMR Amazon yang didukung minimum (5.x, 6.x, 7.x)
	g3s.xlarge	emr-5.19.0, emr-6.0.0, emr-7.0.0
	g4dn.xlarge	emr-5.30.0, emr-6.0.0, emr-7.0.0
	g4dn.2xlarge	emr-5.30.0, emr-6.0.0, emr-7.0.0
	g4dn.4xlarge	emr-5.30.0, emr-6.0.0, emr-7.0.0
	g4dn.8xlarge	emr-5.30.0, emr-6.0.0, emr-7.0.0
	g4dn.12xlarge	emr-5.30.0, emr-6.0.0, emr-7.0.0
	g4dn.16xlarge	emr-5.30.0, emr-6.0.0, emr-7.0.0
	g5.xlarge	emr-5.36.1, emr-6.9.0, emr-7.0.0
	g5.2xlarge	emr-5.36.1, emr-6.9.0, emr-7.0.0
	g5.4xlarge	emr-5.36.1, emr-6.9.0, emr-7.0.0
	g5.8xlarge	emr-5.36.1, emr-6.9.0, emr-7.0.0
	g5.12xlarge	emr-5.36.1, emr-6.9.0, emr-7.0.0

Kelas Instans	Tipe instans	Versi EMR Amazon yang didukung minimum (5.x, 6.x, 7.x)
	g5.16xlarge	emr-5.36.1, emr-6.9.0, emr-7.0.0
	g5.24xlarge	emr-5.36.1, emr-6.9.0, emr-7.0.0
	g5.48xlarge	emr-5.36.1, emr-6.9.0, emr-7.0.0
	p2.xlarge	emr-5.10.0, emr-6.0.0, emr-7.0.0
	p2.8xlarge	emr-5.10.0, emr-6.0.0, emr-7.0.0
	p2.16xlarge	emr-5.10.0, emr-6.0.0, emr-7.0.0
	p3.2xlarge	emr-5.10.0, emr-6.0.0, emr-7.0.0
	p3.8xlarge	emr-5.10.0, emr-6.0.0, emr-7.0.0
	p3.16xlarge	emr-5.10.0, emr-6.0.0, emr-7.0.0
	p5.48xlarge	emr-6.14.0, emr-7.0.0
Memori Dioptimalkan	r5.xlarge	emr-5.13.0, emr-6.0.0, emr-7.0.0
	r5.2xlarge	emr-5.13.0, emr-6.0.0, emr-7.0.0

Kelas Instans	Tipe instans	Versi EMR Amazon yang didukung minimum (5.x, 6.x, 7.x)
	r5.4xlarge	emr-5.13.0, emr-6.0.0, emr-7.0.0
	r5.8xlarge	emr-5.13.0, emr-6.0.0, emr-7.0.0
	r5.12xlarge	emr-5.13.0, emr-6.0.0, emr-7.0.0
	r5.16xlarge	emr-5.13.0, emr-6.0.0, emr-7.0.0
	r5.24xlarge	emr-5.13.0, emr-6.0.0, emr-7.0.0
	r5a.xlarge	emr-5.20.0, emr-6.0.0, emr-7.0.0
	r5a.2xlarge	emr-5.20.0, emr-6.0.0, emr-7.0.0
	r5a.4xlarge	emr-5.20.0, emr-6.0.0, emr-7.0.0
	r5a.8xlarge	emr-5.20.0, emr-6.0.0, emr-7.0.0
	r5a.12xlarge	emr-5.20.0, emr-6.0.0, emr-7.0.0
	r5a.16xlarge	emr-5.20.0, emr-6.0.0, emr-7.0.0
	r5a.24xlarge	emr-5.20.0, emr-6.0.0, emr-7.0.0

Kelas Instans	Tipe instans	Versi EMR Amazon yang didukung minimum (5.x, 6.x, 7.x)
	r5ad.xlarge	emr-5.20.0, emr-6.0.0, emr-7.0.0
	r5ad.2xlarge	emr-5.20.0, emr-6.0.0, emr-7.0.0
	r5ad.4xlarge	emr-5.20.0, emr-6.0.0, emr-7.0.0
	r5ad.8xlarge	emr-5.33.0, emr-6.3.0, emr-7.0.0
	r5ad.12xlarge	emr-5.20.0, emr-6.0.0, emr-7.0.0
	r5ad.16xlarge	emr-5.33.0, emr-6.3.0, emr-7.0.0
	r5ad.24xlarge	emr-5.20.0, emr-6.0.0, emr-7.0.0
	r5b.xlarge	emr-5.33.0, emr-6.3.0, emr-7.0.0
	r5b.2xlarge	emr-5.33.0, emr-6.3.0, emr-7.0.0
	r5b.4xlarge	emr-5.33.0, emr-6.3.0, emr-7.0.0
	r5b.8xlarge	emr-5.33.0, emr-6.3.0, emr-7.0.0
	r5b.12xlarge	emr-5.33.0, emr-6.3.0, emr-7.0.0

Kelas Instans	Tipe instans	Versi EMR Amazon yang didukung minimum (5.x, 6.x, 7.x)
	r5b.16xlarge	emr-5.33.0, emr-6.3.0, emr-7.0.0
	r5b.24xlarge	emr-5.33.0, emr-6.3.0, emr-7.0.0
	r5d.xlarge	emr-5.13.0, emr-6.0.0, emr-7.0.0
	r5d.2xlarge	emr-5.13.0, emr-6.0.0, emr-7.0.0
	r5d.4xlarge	emr-5.13.0, emr-6.0.0, emr-7.0.0
	r5d.8xlarge	emr-5.13.0, emr-6.0.0, emr-7.0.0
	r5d.12xlarge	emr-5.13.0, emr-6.0.0, emr-7.0.0
	r5d.16xlarge	emr-5.13.0, emr-6.0.0, emr-7.0.0
	r5d.24xlarge	emr-5.13.0, emr-6.0.0, emr-7.0.0
	r5dn.xlarge	emr-5.31.0, emr-6.1.0, emr-7.0.0
	r5dn.2xlarge	emr-5.31.0, emr-6.1.0, emr-7.0.0
	r5dn.4xlarge	emr-5.31.0, emr-6.1.0, emr-7.0.0

Kelas Instans	Tipe instans	Versi EMR Amazon yang didukung minimum (5.x, 6.x, 7.x)
	r5dn.8xlarge	emr-5.31.0, emr-6.1.0, emr-7.0.0
	r5dn.12xlarge	emr-5.31.0, emr-6.1.0, emr-7.0.0
	r5dn.16xlarge	emr-5.31.0, emr-6.1.0, emr-7.0.0
	r5dn.24xlarge	emr-5.31.0, emr-6.1.0, emr-7.0.0
	r5n.xlarge	emr-5.31.0, emr-6.1.0, emr-7.0.0
	r5n.2xlarge	emr-5.31.0, emr-6.1.0, emr-7.0.0
	r5n.4xlarge	emr-5.31.0, emr-6.1.0, emr-7.0.0
	r5n.8xlarge	emr-5.31.0, emr-6.1.0, emr-7.0.0
	r5n.12xlarge	emr-5.31.0, emr-6.1.0, emr-7.0.0
	r5n.16xlarge	emr-5.31.0, emr-6.1.0, emr-7.0.0
	r5n.24xlarge	emr-5.31.0, emr-6.1.0, emr-7.0.0
	r6a.xlarge	emr-5.36.1, emr-6.8.0, emr-7.0.0

Kelas Instans	Tipe instans	Versi EMR Amazon yang didukung minimum (5.x, 6.x, 7.x)
	r6a.2xlarge	emr-5.36.1, emr-6.8.0, emr-7.0.0
	r6a.4xlarge	emr-5.36.1, emr-6.8.0, emr-7.0.0
	r6a.8xlarge	emr-5.36.1, emr-6.8.0, emr-7.0.0
	r6a.12xlarge	emr-5.36.1, emr-6.8.0, emr-7.0.0
	r6a.16xlarge	emr-5.36.1, emr-6.8.0, emr-7.0.0
	r6a.24xlarge	emr-5.36.1, emr-6.8.0, emr-7.0.0
	r6a.32xlarge	emr-5.36.1, emr-6.8.0, emr-7.0.0
	r6a.48xlarge	emr-5.36.1, emr-6.8.0, emr-7.0.0
	r6g.xlarge	emr-5.31.0, emr-6.1.0, emr-7.0.0
	r6g.2xlarge	emr-5.31.0, emr-6.1.0, emr-7.0.0
	r6g.4xlarge	emr-5.31.0, emr-6.1.0, emr-7.0.0
	r6g.8xlarge	emr-5.31.0, emr-6.1.0, emr-7.0.0

Kelas Instans	Tipe instans	Versi EMR Amazon yang didukung minimum (5.x, 6.x, 7.x)
	r6g.12xlarge	emr-5.31.0, emr-6.1.0, emr-7.0.0
	r6g.16xlarge	emr-5.31.0, emr-6.1.0, emr-7.0.0
	r6gd.xlarge	emr-5.33.0, emr-6.3.0, emr-7.0.0
	r6gd.2xlarge	emr-5.33.0, emr-6.3.0, emr-7.0.0
	r6gd.4xlarge	emr-5.33.0, emr-6.3.0, emr-7.0.0
	r6gd.8xlarge	emr-5.33.0, emr-6.3.0, emr-7.0.0
	r6gd.12xlarge	emr-5.33.0, emr-6.3.0, emr-7.0.0
	r6gd.16xlarge	emr-5.33.0, emr-6.3.0, emr-7.0.0
	r6i.xlarge	emr-5.35.0, emr-6.6.0
	r6i.2xlarge	emr-5.35.0, emr-6.6.0
	r6i.4xlarge	emr-5.35.0, emr-6.6.0
	r6i.8xlarge	emr-5.35.0, emr-6.6.0
	r6i.12xlarge	emr-5.35.0, emr-6.6.0
	r6i.16xlarge	emr-5.35.0, emr-6.6.0

Kelas Instans	Tipe instans	Versi EMR Amazon yang didukung minimum (5.x, 6.x, 7.x)
	r6i.24xlarge	emr-5.35.0, emr-6.6.0
	r6i.32xlarge	emr-5.35.0, emr-6.6.0
	r6id.xlarge	emr-5.36.1, emr-6.8.0
	r6id.2xlarge	emr-5.36.1, emr-6.8.0
	r6id.4xlarge	emr-5.36.1, emr-6.8.0
	r6id.8xlarge	emr-5.36.1, emr-6.8.0
	r6id.12xlarge	emr-5.36.1, emr-6.8.0
	r6id.16xlarge	emr-5.36.1, emr-6.8.0
	r6id.24xlarge	emr-5.36.1, emr-6.8.0
	r6id.32xlarge	emr-5.36.1, emr-6.8.0
	r6idn.xlarge	emr-5.36.1, emr-6.10.0
	r6idn.2xlarge	emr-5.36.1, emr-6.10.0
	r6idn.4xlarge	emr-5.36.1, emr-6.10.0
	r6idn.8xlarge	emr-5.36.1, emr-6.10.0
	r6idn.12xlarge	emr-5.36.1, emr-6.10.0
	r6idn.16xlarge	emr-5.36.1, emr-6.10.0
	r6idn.24xlarge	emr-5.36.1, emr-6.10.0
	r6idn.32xlarge	emr-5.36.1, emr-6.10.0
	r6in.xlarge	emr-5.36.1, emr-6.10.0

Kelas Instans	Tipe instans	Versi EMR Amazon yang didukung minimum (5.x, 6.x, 7.x)
	r6in.2xlarge	emr-5.36.1, emr-6.10.0
	r6in.4xlarge	emr-5.36.1, emr-6.10.0
	r6in.8xlarge	emr-5.36.1, emr-6.10.0
	r6in.12xlarge	emr-5.36.1, emr-6.10.0
	r6in.16xlarge	emr-5.36.1, emr-6.10.0
	r6in.24xlarge	emr-5.36.1, emr-6.10.0
	r6in.32xlarge	emr-5.36.1, emr-6.10.0
	r7a.xlarge	emr-5.36.1, emr-6.10.0
	r7a.2xlarge	emr-5.36.1, emr-6.10.0
	r7a.4xlarge	emr-5.36.1, emr-6.10.0
	r7a.8xlarge	emr-5.36.1, emr-6.10.0
	r7a.12xlarge	emr-5.36.1, emr-6.10.0
	r7a.16xlarge	emr-5.36.1, emr-6.10.0
	r7a.24xlarge	emr-5.36.1, emr-6.10.0
	r7a.32xlarge	emr-5.36.1, emr-6.10.0
	r7a.48xlarge	emr-5.36.1, emr-6.10.0
	r7g.xlarge	emr-5.36.1, emr-6.10.0, emr-7.0.0
	r7g.2xlarge	emr-5.36.1, emr-6.10.0, emr-7.0.0

Kelas Instans	Tipe instans	Versi EMR Amazon yang didukung minimum (5.x, 6.x, 7.x)
	r7g.4xlarge	emr-5.36.1, emr-6.10.0, emr-7.0.0
	r7g.8xlarge	emr-5.36.1, emr-6.10.0, emr-7.0.0
	r7g.12xlarge	emr-5.36.1, emr-6.10.0, emr-7.0.0
	r7g.16xlarge	emr-5.36.1, emr-6.10.0, emr-7.0.0
	r7gd.xlarge	emr-5.36.1, emr-6.10.0, emr-7.0.0
	r7gd.2xlarge	emr-5.36.1, emr-6.10.0, emr-7.0.0
	r7gd.4xlarge	emr-5.36.1, emr-6.10.0, emr-7.0.0
	r7gd.8xlarge	emr-5.36.1, emr-6.10.0, emr-7.0.0
	r7gd.12xlarge	emr-5.36.1, emr-6.10.0, emr-7.0.0
	r7gd.16xlarge	emr-5.36.1, emr-6.10.0, emr-7.0.0
	r7i.xlarge	emr-5.36.1, emr-6.10.0
	r7i.2xlarge	emr-5.36.1, emr-6.10.0
	r7i.4xlarge	emr-5.36.1, emr-6.10.0

Kelas Instans	Tipe instans	Versi EMR Amazon yang didukung minimum (5.x, 6.x, 7.x)
	r7i.8xlarge	emr-5.36.1, emr-6.10.0
	r7i.12xlarge	emr-5.36.1, emr-6.10.0
	r7i.16xlarge	emr-5.36.1, emr-6.10.0
	r7i.24xlarge	emr-5.36.1, emr-6.10.0
	r7i.48xlarge	emr-5.36.1, emr-6.10.0
	r7iz.xlarge	emr-5.36.1, emr-6.10.0
	r7iz.2xlarge	emr-5.36.1, emr-6.10.0
	r7iz.4xlarge	emr-5.36.1, emr-6.10.0
	r7iz.8xlarge	emr-5.36.1, emr-6.10.0
	r7iz.12xlarge	emr-5.36.1, emr-6.10.0
	r7iz.16xlarge	emr-5.36.1, emr-6.10.0
	r7iz.32xlarge	emr-5.36.1, emr-6.10.0
	x1.16xlarge	emr-5.36.1, emr-6.9.0, emr-7.0.0
	x1.32xlarge	emr-5.36.1, emr-6.9.0, emr-7.0.0
	x1e.xlarge	emr-5.36.1, emr-6.10.0, emr-7.0.0
	x1e.2xlarge	emr-5.36.1, emr-6.10.0, emr-7.0.0

Kelas Instans	Tipe instans	Versi EMR Amazon yang didukung minimum (5.x, 6.x, 7.x)
	x1e.4xlarge	emr-5.36.1, emr-6.10.0, emr-7.0.0
	x1e.8xlarge	emr-5.36.1, emr-6.10.0, emr-7.0.0
	x1e.16xlarge	emr-5.36.1, emr-6.10.0, emr-7.0.0
	x1e.32xlarge	emr-5.36.1, emr-6.10.0, emr-7.0.0
	x2gd.xlarge	emr-5.36.1, emr-6.10.0, emr-7.0.0
	x2gd.2xlarge	emr-5.36.1, emr-6.10.0, emr-7.0.0
	x2gd.4xlarge	emr-5.36.1, emr-6.10.0, emr-7.0.0
	x2gd.8xlarge	emr-5.36.1, emr-6.10.0, emr-7.0.0
	x2gd.12xlarge	emr-5.36.1, emr-6.10.0, emr-7.0.0
	x2gd.16xlarge	emr-5.36.1, emr-6.10.0, emr-7.0.0
	x2idn.16xlarge	emr-5.36.1, emr-6.9.0
	x2idn.24xlarge	emr-5.36.1, emr-6.9.0
	x2idn.32xlarge	emr-5.36.1, emr-6.9.0

Kelas Instans	Tipe instans	Versi EMR Amazon yang didukung minimum (5.x, 6.x, 7.x)
	x2iedn.xlarge	emr-5.36.1, emr-6.9.0
	x2iedn.2xlarge	emr-5.36.1, emr-6.9.0
	x2iedn.4xlarge	emr-5.36.1, emr-6.9.0
	x2iedn.8xlarge	emr-5.36.1, emr-6.9.0
	x2iedn.16xlarge	emr-5.36.1, emr-6.9.0
	x2iedn.24xlarge	emr-5.36.1, emr-6.9.0
	x2iedn.32xlarge	emr-5.36.1, emr-6.9.0
	z1d.xlarge	emr-5.17.0, emr-6.0.0, emr-7.0.0
	z1d.2xlarge	emr-5.17.0, emr-6.0.0, emr-7.0.0
	z1d.3xlarge	emr-5.17.0, emr-6.0.0, emr-7.0.0
	z1d.6xlarge	emr-5.17.0, emr-6.0.0, emr-7.0.0
	z1d.12xlarge	emr-5.17.0, emr-6.0.0, emr-7.0.0
Penyimpanan Dioptimalkan	d3.xlarge	emr-5.33.0, emr-6.3.0, emr-7.0.0
	d3.2xlarge	emr-5.33.0, emr-6.3.0, emr-7.0.0

Kelas Instans	Tipe instans	Versi EMR Amazon yang didukung minimum (5.x, 6.x, 7.x)
	d3.4xlarge	emr-5.33.0, emr-6.3.0, emr-7.0.0
	d3.8xlarge	emr-5.33.0, emr-6.3.0, emr-7.0.0
	d3en.xlarge	emr-5.33.0, emr-6.3.0, emr-7.0.0
	d3en.2xlarge	emr-5.33.0, emr-6.3.0, emr-7.0.0
	d3en.4xlarge	emr-5.33.0, emr-6.3.0, emr-7.0.0
	d3en.6xlarge	emr-5.33.0, emr-6.3.0, emr-7.0.0
	d3en.8xlarge	emr-5.33.0, emr-6.3.0, emr-7.0.0
	d3en.12xlarge	emr-5.33.0, emr-6.3.0, emr-7.0.0
	h1.2xlarge	emr-5.17.0, emr-6.0.0, emr-7.0.0
	h1.4xlarge	emr-5.17.0, emr-6.0.0, emr-7.0.0
	h1.8xlarge	emr-5.17.0, emr-6.0.0, emr-7.0.0
	h1.16xlarge	emr-5.17.0, emr-6.0.0, emr-7.0.0

Kelas Instans	Tipe instans	Versi EMR Amazon yang didukung minimum (5.x, 6.x, 7.x)
	i3.xlarge	emr-5.9.0, emr-6.0.0, emr-7.0.0
	i3.2xlarge	emr-5.9.0, emr-6.0.0, emr-7.0.0
	i3.4xlarge	emr-5.9.0, emr-6.0.0, emr-7.0.0
	i3.8xlarge	emr-5.9.0, emr-6.0.0, emr-7.0.0
	i3.16xlarge	emr-5.9.0, emr-6.0.0, emr-7.0.0
	i3en.xlarge	emr-5.25.0, emr-6.0.0, emr-7.0.0
	i3en.2xlarge	emr-5.25.0, emr-6.0.0, emr-7.0.0
	i3en.3xlarge	emr-5.25.0, emr-6.0.0, emr-7.0.0
	i3en.6xlarge	emr-5.25.0, emr-6.0.0, emr-7.0.0
	i3en.12xlarge	emr-5.25.0, emr-6.0.0, emr-7.0.0
	i3en.24xlarge	emr-5.25.0, emr-6.0.0, emr-7.0.0
	i4g.xlarge	emr-5.36.1, emr-6.10.0, emr-7.0.0

Kelas Instans	Tipe instans	Versi EMR Amazon yang didukung minimum (5.x, 6.x, 7.x)
	i4g.2xlarge	emr-5.36.1, emr-6.10.0, emr-7.0.0
	i4g.4xlarge	emr-5.36.1, emr-6.10.0, emr-7.0.0
	i4g.8xlarge	emr-5.36.1, emr-6.10.0, emr-7.0.0
	i4g.16xlarge	emr-5.36.1, emr-6.10.0, emr-7.0.0
	i4i.xlarge	emr-5.36.1, emr-6.8.0
	i4i.2xlarge	emr-5.36.1, emr-6.8.0
	i4i.4xlarge	emr-5.36.1, emr-6.8.0
	i4i.8xlarge	emr-5.36.1, emr-6.8.0
	i4i.12xlarge	emr-5.36.1, emr-6.10.0
	i4i.16xlarge	emr-5.36.1, emr-6.8.0
	i4i.24xlarge	emr-5.36.1, emr-6.10.0
	i4i.32xlarge	emr-5.36.1, emr-6.8.0
	im4gn.xlarge	emr-5.35.0, emr-6.6.0, emr-7.0.0
	im4gn.2xlarge	emr-5.35.0, emr-6.6.0, emr-7.0.0
	im4gn.4xlarge	emr-5.35.0, emr-6.6.0, emr-7.0.0

Kelas Instans	Tipe instans	Versi EMR Amazon yang didukung minimum (5.x, 6.x, 7.x)
	im4gn.8xlarge	emr-5.35.0, emr-6.6.0, emr-7.0.0
	im4gn.16xlarge	emr-5.35.0, emr-6.6.0, emr-7.0.0
	is4gen.xlarge	emr-5.35.0, emr-6.6.0, emr-7.0.0
	is4gen.2xlarge	emr-5.35.0, emr-6.6.0, emr-7.0.0
	is4gen.4xlarge	emr-5.35.0, emr-6.6.0, emr-7.0.0
	is4gen.8xlarge	emr-5.35.0, emr-6.6.0, emr-7.0.0

US East (Ohio) - us-east-2

Kelas Instans	Tipe instans	Versi EMR Amazon yang didukung minimum (5.x, 6.x, 7.x)
Tujuan Umum	m5.xlarge	emr-5.13.0, emr-6.0.0, emr-7.0.0
	m5.2xlarge	emr-5.13.0, emr-6.0.0, emr-7.0.0
	m5.4xlarge	emr-5.13.0, emr-6.0.0, emr-7.0.0

Kelas Instans	Tipe instans	Versi EMR Amazon yang didukung minimum (5.x, 6.x, 7.x)
	m5.8xlarge	emr-5.13.0, emr-6.0.0, emr-7.0.0
	m5.12xlarge	emr-5.13.0, emr-6.0.0, emr-7.0.0
	m5.16xlarge	emr-5.13.0, emr-6.0.0, emr-7.0.0
	m5.24xlarge	emr-5.13.0, emr-6.0.0, emr-7.0.0
	m5a.xlarge	emr-5.20.0, emr-6.0.0, emr-7.0.0
	m5a.2xlarge	emr-5.20.0, emr-6.0.0, emr-7.0.0
	m5a.4xlarge	emr-5.20.0, emr-6.0.0, emr-7.0.0
	m5a.8xlarge	emr-5.20.0, emr-6.0.0, emr-7.0.0
	m5a.12xlarge	emr-5.20.0, emr-6.0.0, emr-7.0.0
	m5a.16xlarge	emr-5.20.0, emr-6.0.0, emr-7.0.0
	m5a.24xlarge	emr-5.20.0, emr-6.0.0, emr-7.0.0
	m5ad.xlarge	emr-5.20.0, emr-6.0.0, emr-7.0.0

Kelas Instans	Tipe instans	Versi EMR Amazon yang didukung minimum (5.x, 6.x, 7.x)
	m5ad.2xlarge	emr-5.20.0, emr-6.0.0, emr-7.0.0
	m5ad.4xlarge	emr-5.20.0, emr-6.0.0, emr-7.0.0
	m5ad.8xlarge	emr-5.33.0, emr-6.3.0, emr-7.0.0
	m5ad.12xlarge	emr-5.20.0, emr-6.0.0, emr-7.0.0
	m5ad.16xlarge	emr-5.33.0, emr-6.3.0, emr-7.0.0
	m5ad.24xlarge	emr-5.20.0, emr-6.0.0, emr-7.0.0
	m5d.xlarge	emr-5.13.0, emr-6.0.0, emr-7.0.0
	m5d.2xlarge	emr-5.13.0, emr-6.0.0, emr-7.0.0
	m5d.4xlarge	emr-5.13.0, emr-6.0.0, emr-7.0.0
	m5d.8xlarge	emr-5.13.0, emr-6.0.0, emr-7.0.0
	m5d.12xlarge	emr-5.13.0, emr-6.0.0, emr-7.0.0
	m5d.16xlarge	emr-5.13.0, emr-6.0.0, emr-7.0.0

Kelas Instans	Tipe instans	Versi EMR Amazon yang didukung minimum (5.x, 6.x, 7.x)
	m5d.24xlarge	emr-5.13.0, emr-6.0.0, emr-7.0.0
	m5dn.xlarge	emr-5.31.0, emr-6.1.0, emr-7.0.0
	m5dn.2xlarge	emr-5.31.0, emr-6.1.0, emr-7.0.0
	m5dn.4xlarge	emr-5.31.0, emr-6.1.0, emr-7.0.0
	m5dn.8xlarge	emr-5.31.0, emr-6.1.0, emr-7.0.0
	m5dn.12xlarge	emr-5.31.0, emr-6.1.0, emr-7.0.0
	m5dn.16xlarge	emr-5.31.0, emr-6.1.0, emr-7.0.0
	m5dn.24xlarge	emr-5.31.0, emr-6.1.0, emr-7.0.0
	m5n.xlarge	emr-5.31.0, emr-6.1.0, emr-7.0.0
	m5n.2xlarge	emr-5.31.0, emr-6.1.0, emr-7.0.0
	m5n.4xlarge	emr-5.31.0, emr-6.1.0, emr-7.0.0
	m5n.8xlarge	emr-5.31.0, emr-6.1.0, emr-7.0.0

Kelas Instans	Tipe instans	Versi EMR Amazon yang didukung minimum (5.x, 6.x, 7.x)
	m5n.12xlarge	emr-5.31.0, emr-6.1.0, emr-7.0.0
	m5n.16xlarge	emr-5.31.0, emr-6.1.0, emr-7.0.0
	m5n.24xlarge	emr-5.31.0, emr-6.1.0, emr-7.0.0
	m5zn.xlarge	emr-5.33.0, emr-6.3.0, emr-7.0.0
	m5zn.2xlarge	emr-5.33.0, emr-6.3.0, emr-7.0.0
	m5zn.3xlarge	emr-5.33.0, emr-6.3.0, emr-7.0.0
	m5zn.6xlarge	emr-5.33.0, emr-6.3.0, emr-7.0.0
	m5zn.12xlarge	emr-5.33.0, emr-6.3.0, emr-7.0.0
	m6a.xlarge	emr-5.36.1, emr-6.8.0, emr-7.0.0
	m6a.2xlarge	emr-5.36.1, emr-6.8.0, emr-7.0.0
	m6a.4xlarge	emr-5.36.1, emr-6.8.0, emr-7.0.0
	m6a.8xlarge	emr-5.36.1, emr-6.8.0, emr-7.0.0

Kelas Instans	Tipe instans	Versi EMR Amazon yang didukung minimum (5.x, 6.x, 7.x)
	m6a.12xlarge	emr-5.36.1, emr-6.8.0, emr-7.0.0
	m6a.16xlarge	emr-5.36.1, emr-6.8.0, emr-7.0.0
	m6a.24xlarge	emr-5.36.1, emr-6.8.0, emr-7.0.0
	m6a.32xlarge	emr-5.36.1, emr-6.8.0, emr-7.0.0
	m6a.48xlarge	emr-5.36.1, emr-6.8.0, emr-7.0.0
	m6g.xlarge	emr-5.30.0, emr-6.1.0, emr-7.0.0
	m6g.2xlarge	emr-5.30.0, emr-6.1.0, emr-7.0.0
	m6g.4xlarge	emr-5.30.0, emr-6.1.0, emr-7.0.0
	m6g.8xlarge	emr-5.30.0, emr-6.1.0, emr-7.0.0
	m6g.12xlarge	emr-5.30.0, emr-6.1.0, emr-7.0.0
	m6g.16xlarge	emr-5.30.0, emr-6.1.0, emr-7.0.0
	m6gd.xlarge	emr-5.33.0, emr-6.3.0, emr-7.0.0

Kelas Instans	Tipe instans	Versi EMR Amazon yang didukung minimum (5.x, 6.x, 7.x)
	m6gd.2xlarge	emr-5.33.0, emr-6.3.0, emr-7.0.0
	m6gd.4xlarge	emr-5.33.0, emr-6.3.0, emr-7.0.0
	m6gd.8xlarge	emr-5.33.0, emr-6.3.0, emr-7.0.0
	m6gd.12xlarge	emr-5.33.0, emr-6.3.0, emr-7.0.0
	m6gd.16xlarge	emr-5.33.0, emr-6.3.0, emr-7.0.0
	m6i.xlarge	emr-5.35.0, emr-6.6.0
	m6i.2xlarge	emr-5.35.0, emr-6.6.0
	m6i.4xlarge	emr-5.35.0, emr-6.6.0
	m6i.8xlarge	emr-5.35.0, emr-6.6.0
	m6i.12xlarge	emr-5.35.0, emr-6.6.0
	m6i.16xlarge	emr-5.35.0, emr-6.6.0
	m6i.24xlarge	emr-5.35.0, emr-6.6.0
	m6i.32xlarge	emr-5.35.0, emr-6.6.0
	m6id.xlarge	emr-5.36.1, emr-6.8.0
	m6id.2xlarge	emr-5.36.1, emr-6.8.0
	m6id.4xlarge	emr-5.36.1, emr-6.8.0

Kelas Instans	Tipe instans	Versi EMR Amazon yang didukung minimum (5.x, 6.x, 7.x)
	m6id.8xlarge	emr-5.36.1, emr-6.8.0
	m6id.12xlarge	emr-5.36.1, emr-6.8.0
	m6id.16xlarge	emr-5.36.1, emr-6.8.0
	m6id.24xlarge	emr-5.36.1, emr-6.8.0
	m6id.32xlarge	emr-5.36.1, emr-6.8.0
	m6idn.xlarge	emr-5.36.1, emr-6.10.0
	m6idn.2xlarge	emr-5.36.1, emr-6.10.0
	m6idn.4xlarge	emr-5.36.1, emr-6.10.0
	m6idn.8xlarge	emr-5.36.1, emr-6.10.0
	m6idn.12xlarge	emr-5.36.1, emr-6.10.0
	m6idn.16xlarge	emr-5.36.1, emr-6.10.0
	m6idn.24xlarge	emr-5.36.1, emr-6.10.0
	m6idn.32xlarge	emr-5.36.1, emr-6.10.0
	m6in.xlarge	emr-5.36.1, emr-6.10.0
	m6in.2xlarge	emr-5.36.1, emr-6.10.0
	m6in.4xlarge	emr-5.36.1, emr-6.10.0
	m6in.8xlarge	emr-5.36.1, emr-6.10.0
	m6in.12xlarge	emr-5.36.1, emr-6.10.0
	m6in.16xlarge	emr-5.36.1, emr-6.10.0

Kelas Instans	Tipe instans	Versi EMR Amazon yang didukung minimum (5.x, 6.x, 7.x)
	m6in.24xlarge	emr-5.36.1, emr-6.10.0
	m6in.32xlarge	emr-5.36.1, emr-6.10.0
	m7a.xlarge	emr-5.36.1, emr-6.10.0
	m7a.2xlarge	emr-5.36.1, emr-6.10.0
	m7a.4xlarge	emr-5.36.1, emr-6.10.0
	m7a.8xlarge	emr-5.36.1, emr-6.10.0
	m7a.12xlarge	emr-5.36.1, emr-6.10.0
	m7a.16xlarge	emr-5.36.1, emr-6.10.0
	m7a.24xlarge	emr-5.36.1, emr-6.10.0
	m7a.32xlarge	emr-5.36.1, emr-6.10.0
	m7a.48xlarge	emr-5.36.1, emr-6.10.0
	m7g.xlarge	emr-5.36.1, emr-6.10.0, emr-7.0.0
	m7g.2xlarge	emr-5.36.1, emr-6.10.0, emr-7.0.0
	m7g.4xlarge	emr-5.36.1, emr-6.10.0, emr-7.0.0
	m7g.8xlarge	emr-5.36.1, emr-6.10.0, emr-7.0.0
	m7g.12xlarge	emr-5.36.1, emr-6.10.0, emr-7.0.0

Kelas Instans	Tipe instans	Versi EMR Amazon yang didukung minimum (5.x, 6.x, 7.x)
	m7g.16xlarge	emr-5.36.1, emr-6.10.0, emr-7.0.0
	m7gd.xlarge	emr-5.36.1, emr-6.10.0, emr-7.0.0
	m7gd.2xlarge	emr-5.36.1, emr-6.10.0, emr-7.0.0
	m7gd.4xlarge	emr-5.36.1, emr-6.10.0, emr-7.0.0
	m7gd.8xlarge	emr-5.36.1, emr-6.10.0, emr-7.0.0
	m7gd.12xlarge	emr-5.36.1, emr-6.10.0, emr-7.0.0
	m7gd.16xlarge	emr-5.36.1, emr-6.10.0, emr-7.0.0
	m7i.xlarge	emr-5.36.1, emr-6.10.0
	m7i.2xlarge	emr-5.36.1, emr-6.10.0
	m7i.4xlarge	emr-5.36.1, emr-6.10.0
	m7i.8xlarge	emr-5.36.1, emr-6.10.0
	m7i.12xlarge	emr-5.36.1, emr-6.10.0
	m7i.16xlarge	emr-5.36.1, emr-6.10.0
	m7i.24xlarge	emr-5.36.1, emr-6.10.0
	m7i.48xlarge	emr-5.36.1, emr-6.10.0

Kelas Instans	Tipe instans	Versi EMR Amazon yang didukung minimum (5.x, 6.x, 7.x)
	m7i-flex.xlarge	emr-5.36.1, emr-6.10.0
	m7i-flex.2xlarge	emr-5.36.1, emr-6.10.0
	m7i-flex.4xlarge	emr-5.36.1, emr-6.10.0
	m7i-flex.8xlarge	emr-5.36.1, emr-6.10.0
Komputasi Dioptimalkan	c5.xlarge	emr-5.13.0, emr-6.0.0, emr-7.0.0
	c5.2xlarge	emr-5.13.0, emr-6.0.0, emr-7.0.0
	c5.4xlarge	emr-5.13.0, emr-6.0.0, emr-7.0.0
	c5.9xlarge	emr-5.13.0, emr-6.0.0, emr-7.0.0
	c5.12xlarge	emr-5.13.0, emr-6.0.0, emr-7.0.0
	c5.18xlarge	emr-5.13.0, emr-6.0.0, emr-7.0.0
	c5.24xlarge	emr-5.13.0, emr-6.0.0, emr-7.0.0
	c5a.xlarge	emr-5.31.0, emr-6.1.0, emr-7.0.0
	c5a.2xlarge	emr-5.31.0, emr-6.1.0, emr-7.0.0

Kelas Instans	Tipe instans	Versi EMR Amazon yang didukung minimum (5.x, 6.x, 7.x)
	c5a.4xlarge	emr-5.31.0, emr-6.1.0, emr-7.0.0
	c5a.8xlarge	emr-5.31.0, emr-6.1.0, emr-7.0.0
	c5a.12xlarge	emr-5.31.0, emr-6.1.0, emr-7.0.0
	c5a.16xlarge	emr-5.31.0, emr-6.1.0, emr-7.0.0
	c5a.24xlarge	emr-5.31.0, emr-6.1.0, emr-7.0.0
	c5ad.xlarge	emr-5.33.0, emr-6.3.0, emr-7.0.0
	c5ad.2xlarge	emr-5.33.0, emr-6.3.0, emr-7.0.0
	c5ad.4xlarge	emr-5.33.0, emr-6.3.0, emr-7.0.0
	c5ad.8xlarge	emr-5.33.0, emr-6.3.0, emr-7.0.0
	c5ad.12xlarge	emr-5.33.0, emr-6.3.0, emr-7.0.0
	c5ad.16xlarge	emr-5.33.0, emr-6.3.0, emr-7.0.0
	c5ad.24xlarge	emr-5.33.0, emr-6.3.0, emr-7.0.0

Kelas Instans	Tipe instans	Versi EMR Amazon yang didukung minimum (5.x, 6.x, 7.x)
	c5d.xlarge	emr-5.13.0, emr-6.0.0, emr-7.0.0
	c5d.2xlarge	emr-5.13.0, emr-6.0.0, emr-7.0.0
	c5d.4xlarge	emr-5.13.0, emr-6.0.0, emr-7.0.0
	c5d.9xlarge	emr-5.13.0, emr-6.0.0, emr-7.0.0
	c5d.12xlarge	emr-5.13.0, emr-6.0.0, emr-7.0.0
	c5d.18xlarge	emr-5.13.0, emr-6.0.0, emr-7.0.0
	c5d.24xlarge	emr-5.13.0, emr-6.0.0, emr-7.0.0
	c5n.xlarge	emr-5.20.0, emr-6.0.0, emr-7.0.0
	c5n.2xlarge	emr-5.20.0, emr-6.0.0, emr-7.0.0
	c5n.4xlarge	emr-5.20.0, emr-6.0.0, emr-7.0.0
	c5n.9xlarge	emr-5.20.0, emr-6.0.0, emr-7.0.0
	c5n.18xlarge	emr-5.20.0, emr-6.0.0, emr-7.0.0

Kelas Instans	Tipe instans	Versi EMR Amazon yang didukung minimum (5.x, 6.x, 7.x)
	c6a.xlarge	emr-5.36.1, emr-6.8.0, emr-7.0.0
	c6a.2xlarge	emr-5.36.1, emr-6.8.0, emr-7.0.0
	c6a.4xlarge	emr-5.36.1, emr-6.8.0, emr-7.0.0
	c6a.8xlarge	emr-5.36.1, emr-6.8.0, emr-7.0.0
	c6a.12xlarge	emr-5.36.1, emr-6.8.0, emr-7.0.0
	c6a.16xlarge	emr-5.36.1, emr-6.8.0, emr-7.0.0
	c6a.24xlarge	emr-5.36.1, emr-6.8.0, emr-7.0.0
	c6a.32xlarge	emr-5.36.1, emr-6.8.0, emr-7.0.0
	c6a.48xlarge	emr-5.36.1, emr-6.8.0, emr-7.0.0
	c6g.xlarge	emr-5.31.0, emr-6.1.0, emr-7.0.0
	c6g.2xlarge	emr-5.31.0, emr-6.1.0, emr-7.0.0
	c6g.4xlarge	emr-5.31.0, emr-6.1.0, emr-7.0.0

Kelas Instans	Tipe instans	Versi EMR Amazon yang didukung minimum (5.x, 6.x, 7.x)
	c6g.8xlarge	emr-5.31.0, emr-6.1.0, emr-7.0.0
	c6g.12xlarge	emr-5.31.0, emr-6.1.0, emr-7.0.0
	c6g.16xlarge	emr-5.31.0, emr-6.1.0, emr-7.0.0
	c6gd.xlarge	emr-5.33.0, emr-6.3.0, emr-7.0.0
	c6gd.2xlarge	emr-5.33.0, emr-6.3.0, emr-7.0.0
	c6gd.4xlarge	emr-5.33.0, emr-6.3.0, emr-7.0.0
	c6gd.8xlarge	emr-5.33.0, emr-6.3.0, emr-7.0.0
	c6gd.12xlarge	emr-5.33.0, emr-6.3.0, emr-7.0.0
	c6gd.16xlarge	emr-5.33.0, emr-6.3.0, emr-7.0.0
	c6gn.xlarge	emr-5.33.0, emr-6.3.0, emr-7.0.0
	c6gn.2xlarge	emr-5.33.0, emr-6.3.0, emr-7.0.0
	c6gn.4xlarge	emr-5.33.0, emr-6.3.0, emr-7.0.0

Kelas Instans	Tipe instans	Versi EMR Amazon yang didukung minimum (5.x, 6.x, 7.x)
	c6gn.8xlarge	emr-5.33.0, emr-6.3.0, emr-7.0.0
	c6gn.12xlarge	emr-5.33.0, emr-6.3.0, emr-7.0.0
	c6gn.16xlarge	emr-5.33.0, emr-6.3.0, emr-7.0.0
	c6i.xlarge	emr-5.35.0, emr-6.6.0
	c6i.2xlarge	emr-5.35.0, emr-6.6.0
	c6i.4xlarge	emr-5.35.0, emr-6.6.0
	c6i.8xlarge	emr-5.35.0, emr-6.6.0
	c6i.12xlarge	emr-5.35.0, emr-6.6.0
	c6i.16xlarge	emr-5.35.0, emr-6.6.0
	c6i.24xlarge	emr-5.35.0, emr-6.6.0
	c6i.32xlarge	emr-5.35.0, emr-6.6.0
	c6id.xlarge	emr-5.36.1, emr-6.8.0
	c6id.2xlarge	emr-5.36.1, emr-6.8.0
	c6id.4xlarge	emr-5.36.1, emr-6.8.0
	c6id.8xlarge	emr-5.36.1, emr-6.8.0
	c6id.12xlarge	emr-5.36.1, emr-6.8.0
	c6id.16xlarge	emr-5.36.1, emr-6.8.0

Kelas Instans	Tipe instans	Versi EMR Amazon yang didukung minimum (5.x, 6.x, 7.x)
	c6id.24xlarge	emr-5.36.1, emr-6.8.0
	c6id.32xlarge	emr-5.36.1, emr-6.8.0
	c6in.xlarge	emr-5.36.1, emr-6.10.0
	c6in.2xlarge	emr-5.36.1, emr-6.10.0
	c6in.4xlarge	emr-5.36.1, emr-6.10.0
	c6in.8xlarge	emr-5.36.1, emr-6.10.0
	c6in.12xlarge	emr-5.36.1, emr-6.10.0
	c6in.16xlarge	emr-5.36.1, emr-6.10.0
	c6in.24xlarge	emr-5.36.1, emr-6.10.0
	c6in.32xlarge	emr-5.36.1, emr-6.10.0
	c7a.xlarge	emr-5.36.1, emr-6.10.0
	c7a.2xlarge	emr-5.36.1, emr-6.10.0
	c7a.4xlarge	emr-5.36.1, emr-6.10.0
	c7a.8xlarge	emr-5.36.1, emr-6.10.0
	c7a.12xlarge	emr-5.36.1, emr-6.10.0
	c7a.16xlarge	emr-5.36.1, emr-6.10.0
	c7a.24xlarge	emr-5.36.1, emr-6.10.0
	c7a.32xlarge	emr-5.36.1, emr-6.10.0
	c7a.48xlarge	emr-5.36.1, emr-6.10.0

Kelas Instans	Tipe instans	Versi EMR Amazon yang didukung minimum (5.x, 6.x, 7.x)
	c7g.xlarge	emr-5.36.1, emr-6.7.0, emr-7.0.0
	c7g.2xlarge	emr-5.36.1, emr-6.7.0, emr-7.0.0
	c7g.4xlarge	emr-5.36.1, emr-6.7.0, emr-7.0.0
	c7g.8xlarge	emr-5.36.1, emr-6.7.0, emr-7.0.0
	c7g.12xlarge	emr-5.36.1, emr-6.7.0, emr-7.0.0
	c7g.16xlarge	emr-5.36.1, emr-6.7.0, emr-7.0.0
	c7gd.xlarge	emr-5.36.1, emr-6.10.0, emr-7.0.0
	c7gd.2xlarge	emr-5.36.1, emr-6.10.0, emr-7.0.0
	c7gd.4xlarge	emr-5.36.1, emr-6.10.0, emr-7.0.0
	c7gd.8xlarge	emr-5.36.1, emr-6.10.0, emr-7.0.0
	c7gd.12xlarge	emr-5.36.1, emr-6.10.0, emr-7.0.0
	c7gd.16xlarge	emr-5.36.1, emr-6.10.0, emr-7.0.0

Kelas Instans	Tipe instans	Versi EMR Amazon yang didukung minimum (5.x, 6.x, 7.x)
	c7gn.xlarge	emr-5.36.1, emr-6.10.0, emr-7.0.0
	c7gn.2xlarge	emr-5.36.1, emr-6.10.0, emr-7.0.0
	c7gn.4xlarge	emr-5.36.1, emr-6.10.0, emr-7.0.0
	c7gn.8xlarge	emr-5.36.1, emr-6.10.0, emr-7.0.0
	c7gn.12xlarge	emr-5.36.1, emr-6.10.0, emr-7.0.0
	c7gn.16xlarge	emr-5.36.1, emr-6.10.0, emr-7.0.0
	c7i.xlarge	emr-5.36.1, emr-6.10.0
	c7i.2xlarge	emr-5.36.1, emr-6.10.0
	c7i.4xlarge	emr-5.36.1, emr-6.10.0
	c7i.8xlarge	emr-5.36.1, emr-6.10.0
	c7i.12xlarge	emr-5.36.1, emr-6.10.0
	c7i.16xlarge	emr-5.36.1, emr-6.10.0
	c7i.24xlarge	emr-5.36.1, emr-6.10.0
	c7i.48xlarge	emr-5.36.1, emr-6.10.0
Komputasi yang Dipercepat	g3.4xlarge	emr-5.18.0, emr-6.0.0, emr-7.0.0

Kelas Instans	Tipe instans	Versi EMR Amazon yang didukung minimum (5.x, 6.x, 7.x)
	g3.8xlarge	emr-5.18.0, emr-6.0.0, emr-7.0.0
	g3.16xlarge	emr-5.18.0, emr-6.0.0, emr-7.0.0
	g3s.xlarge	emr-5.19.0, emr-6.0.0, emr-7.0.0
	g4dn.xlarge	emr-5.30.0, emr-6.0.0, emr-7.0.0
	g4dn.2xlarge	emr-5.30.0, emr-6.0.0, emr-7.0.0
	g4dn.4xlarge	emr-5.30.0, emr-6.0.0, emr-7.0.0
	g4dn.8xlarge	emr-5.30.0, emr-6.0.0, emr-7.0.0
	g4dn.12xlarge	emr-5.30.0, emr-6.0.0, emr-7.0.0
	g4dn.16xlarge	emr-5.30.0, emr-6.0.0, emr-7.0.0
	g5.xlarge	emr-5.36.1, emr-6.9.0, emr-7.0.0
	g5.2xlarge	emr-5.36.1, emr-6.9.0, emr-7.0.0
	g5.4xlarge	emr-5.36.1, emr-6.9.0, emr-7.0.0

Kelas Instans	Tipe instans	Versi EMR Amazon yang didukung minimum (5.x, 6.x, 7.x)
	g5.8xlarge	emr-5.36.1, emr-6.9.0, emr-7.0.0
	g5.12xlarge	emr-5.36.1, emr-6.9.0, emr-7.0.0
	g5.16xlarge	emr-5.36.1, emr-6.9.0, emr-7.0.0
	g5.24xlarge	emr-5.36.1, emr-6.9.0, emr-7.0.0
	g5.48xlarge	emr-5.36.1, emr-6.9.0, emr-7.0.0
	p2.xlarge	emr-5.10.0, emr-6.0.0, emr-7.0.0
	p2.8xlarge	emr-5.10.0, emr-6.0.0, emr-7.0.0
	p2.16xlarge	emr-5.10.0, emr-6.0.0, emr-7.0.0
	p3.2xlarge	emr-5.10.0, emr-6.0.0, emr-7.0.0
	p3.8xlarge	emr-5.10.0, emr-6.0.0, emr-7.0.0
	p3.16xlarge	emr-5.10.0, emr-6.0.0, emr-7.0.0
	p5.48xlarge	emr-6.14.0, emr-7.0.0

Kelas Instans	Tipe instans	Versi EMR Amazon yang didukung minimum (5.x, 6.x, 7.x)
Memori Dioptimalkan	r5.xlarge	emr-5.13.0, emr-6.0.0, emr-7.0.0
	r5.2xlarge	emr-5.13.0, emr-6.0.0, emr-7.0.0
	r5.4xlarge	emr-5.13.0, emr-6.0.0, emr-7.0.0
	r5.8xlarge	emr-5.13.0, emr-6.0.0, emr-7.0.0
	r5.12xlarge	emr-5.13.0, emr-6.0.0, emr-7.0.0
	r5.16xlarge	emr-5.13.0, emr-6.0.0, emr-7.0.0
	r5.24xlarge	emr-5.13.0, emr-6.0.0, emr-7.0.0
	r5a.xlarge	emr-5.20.0, emr-6.0.0, emr-7.0.0
	r5a.2xlarge	emr-5.20.0, emr-6.0.0, emr-7.0.0
	r5a.4xlarge	emr-5.20.0, emr-6.0.0, emr-7.0.0
	r5a.8xlarge	emr-5.20.0, emr-6.0.0, emr-7.0.0
	r5a.12xlarge	emr-5.20.0, emr-6.0.0, emr-7.0.0

Kelas Instans	Tipe instans	Versi EMR Amazon yang didukung minimum (5.x, 6.x, 7.x)
	r5a.16xlarge	emr-5.20.0, emr-6.0.0, emr-7.0.0
	r5a.24xlarge	emr-5.20.0, emr-6.0.0, emr-7.0.0
	r5ad.xlarge	emr-5.20.0, emr-6.0.0, emr-7.0.0
	r5ad.2xlarge	emr-5.20.0, emr-6.0.0, emr-7.0.0
	r5ad.4xlarge	emr-5.20.0, emr-6.0.0, emr-7.0.0
	r5ad.8xlarge	emr-5.33.0, emr-6.3.0, emr-7.0.0
	r5ad.12xlarge	emr-5.20.0, emr-6.0.0, emr-7.0.0
	r5ad.16xlarge	emr-5.33.0, emr-6.3.0, emr-7.0.0
	r5ad.24xlarge	emr-5.20.0, emr-6.0.0, emr-7.0.0
	r5b.xlarge	emr-5.33.0, emr-6.3.0, emr-7.0.0
	r5b.2xlarge	emr-5.33.0, emr-6.3.0, emr-7.0.0
	r5b.4xlarge	emr-5.33.0, emr-6.3.0, emr-7.0.0

Kelas Instans	Tipe instans	Versi EMR Amazon yang didukung minimum (5.x, 6.x, 7.x)
	r5b.8xlarge	emr-5.33.0, emr-6.3.0, emr-7.0.0
	r5b.12xlarge	emr-5.33.0, emr-6.3.0, emr-7.0.0
	r5b.16xlarge	emr-5.33.0, emr-6.3.0, emr-7.0.0
	r5b.24xlarge	emr-5.33.0, emr-6.3.0, emr-7.0.0
	r5d.xlarge	emr-5.13.0, emr-6.0.0, emr-7.0.0
	r5d.2xlarge	emr-5.13.0, emr-6.0.0, emr-7.0.0
	r5d.4xlarge	emr-5.13.0, emr-6.0.0, emr-7.0.0
	r5d.8xlarge	emr-5.13.0, emr-6.0.0, emr-7.0.0
	r5d.12xlarge	emr-5.13.0, emr-6.0.0, emr-7.0.0
	r5d.16xlarge	emr-5.13.0, emr-6.0.0, emr-7.0.0
	r5d.24xlarge	emr-5.13.0, emr-6.0.0, emr-7.0.0
	r5dn.xlarge	emr-5.31.0, emr-6.1.0, emr-7.0.0

Kelas Instans	Tipe instans	Versi EMR Amazon yang didukung minimum (5.x, 6.x, 7.x)
	r5dn.2xlarge	emr-5.31.0, emr-6.1.0, emr-7.0.0
	r5dn.4xlarge	emr-5.31.0, emr-6.1.0, emr-7.0.0
	r5dn.8xlarge	emr-5.31.0, emr-6.1.0, emr-7.0.0
	r5dn.12xlarge	emr-5.31.0, emr-6.1.0, emr-7.0.0
	r5dn.16xlarge	emr-5.31.0, emr-6.1.0, emr-7.0.0
	r5dn.24xlarge	emr-5.31.0, emr-6.1.0, emr-7.0.0
	r5n.xlarge	emr-5.31.0, emr-6.1.0, emr-7.0.0
	r5n.2xlarge	emr-5.31.0, emr-6.1.0, emr-7.0.0
	r5n.4xlarge	emr-5.31.0, emr-6.1.0, emr-7.0.0
	r5n.8xlarge	emr-5.31.0, emr-6.1.0, emr-7.0.0
	r5n.12xlarge	emr-5.31.0, emr-6.1.0, emr-7.0.0
	r5n.16xlarge	emr-5.31.0, emr-6.1.0, emr-7.0.0

Kelas Instans	Tipe instans	Versi EMR Amazon yang didukung minimum (5.x, 6.x, 7.x)
	r5n.24xlarge	emr-5.31.0, emr-6.1.0, emr-7.0.0
	r6a.xlarge	emr-5.36.1, emr-6.8.0, emr-7.0.0
	r6a.2xlarge	emr-5.36.1, emr-6.8.0, emr-7.0.0
	r6a.4xlarge	emr-5.36.1, emr-6.8.0, emr-7.0.0
	r6a.8xlarge	emr-5.36.1, emr-6.8.0, emr-7.0.0
	r6a.12xlarge	emr-5.36.1, emr-6.8.0, emr-7.0.0
	r6a.16xlarge	emr-5.36.1, emr-6.8.0, emr-7.0.0
	r6a.24xlarge	emr-5.36.1, emr-6.8.0, emr-7.0.0
	r6a.32xlarge	emr-5.36.1, emr-6.8.0, emr-7.0.0
	r6a.48xlarge	emr-5.36.1, emr-6.8.0, emr-7.0.0
	r6g.xlarge	emr-5.31.0, emr-6.1.0, emr-7.0.0
	r6g.2xlarge	emr-5.31.0, emr-6.1.0, emr-7.0.0

Kelas Instans	Tipe instans	Versi EMR Amazon yang didukung minimum (5.x, 6.x, 7.x)
	r6g.4xlarge	emr-5.31.0, emr-6.1.0, emr-7.0.0
	r6g.8xlarge	emr-5.31.0, emr-6.1.0, emr-7.0.0
	r6g.12xlarge	emr-5.31.0, emr-6.1.0, emr-7.0.0
	r6g.16xlarge	emr-5.31.0, emr-6.1.0, emr-7.0.0
	r6gd.xlarge	emr-5.33.0, emr-6.3.0, emr-7.0.0
	r6gd.2xlarge	emr-5.33.0, emr-6.3.0, emr-7.0.0
	r6gd.4xlarge	emr-5.33.0, emr-6.3.0, emr-7.0.0
	r6gd.8xlarge	emr-5.33.0, emr-6.3.0, emr-7.0.0
	r6gd.12xlarge	emr-5.33.0, emr-6.3.0, emr-7.0.0
	r6gd.16xlarge	emr-5.33.0, emr-6.3.0, emr-7.0.0
	r6i.xlarge	emr-5.35.0, emr-6.6.0
	r6i.2xlarge	emr-5.35.0, emr-6.6.0
	r6i.4xlarge	emr-5.35.0, emr-6.6.0

Kelas Instans	Tipe instans	Versi EMR Amazon yang didukung minimum (5.x, 6.x, 7.x)
	r6i.8xlarge	emr-5.35.0, emr-6.6.0
	r6i.12xlarge	emr-5.35.0, emr-6.6.0
	r6i.16xlarge	emr-5.35.0, emr-6.6.0
	r6i.24xlarge	emr-5.35.0, emr-6.6.0
	r6i.32xlarge	emr-5.35.0, emr-6.6.0
	r6id.xlarge	emr-5.36.1, emr-6.8.0
	r6id.2xlarge	emr-5.36.1, emr-6.8.0
	r6id.4xlarge	emr-5.36.1, emr-6.8.0
	r6id.8xlarge	emr-5.36.1, emr-6.8.0
	r6id.12xlarge	emr-5.36.1, emr-6.8.0
	r6id.16xlarge	emr-5.36.1, emr-6.8.0
	r6id.24xlarge	emr-5.36.1, emr-6.8.0
	r6id.32xlarge	emr-5.36.1, emr-6.8.0
	r6idn.xlarge	emr-5.36.1, emr-6.10.0
	r6idn.2xlarge	emr-5.36.1, emr-6.10.0
	r6idn.4xlarge	emr-5.36.1, emr-6.10.0
	r6idn.8xlarge	emr-5.36.1, emr-6.10.0
	r6idn.12xlarge	emr-5.36.1, emr-6.10.0
	r6idn.16xlarge	emr-5.36.1, emr-6.10.0

Kelas Instans	Tipe instans	Versi EMR Amazon yang didukung minimum (5.x, 6.x, 7.x)
	r6idn.24xlarge	emr-5.36.1, emr-6.10.0
	r6idn.32xlarge	emr-5.36.1, emr-6.10.0
	r6in.xlarge	emr-5.36.1, emr-6.10.0
	r6in.2xlarge	emr-5.36.1, emr-6.10.0
	r6in.4xlarge	emr-5.36.1, emr-6.10.0
	r6in.8xlarge	emr-5.36.1, emr-6.10.0
	r6in.12xlarge	emr-5.36.1, emr-6.10.0
	r6in.16xlarge	emr-5.36.1, emr-6.10.0
	r6in.24xlarge	emr-5.36.1, emr-6.10.0
	r6in.32xlarge	emr-5.36.1, emr-6.10.0
	r7a.xlarge	emr-5.36.1, emr-6.10.0
	r7a.2xlarge	emr-5.36.1, emr-6.10.0
	r7a.4xlarge	emr-5.36.1, emr-6.10.0
	r7a.8xlarge	emr-5.36.1, emr-6.10.0
	r7a.12xlarge	emr-5.36.1, emr-6.10.0
	r7a.16xlarge	emr-5.36.1, emr-6.10.0
	r7a.24xlarge	emr-5.36.1, emr-6.10.0
	r7a.32xlarge	emr-5.36.1, emr-6.10.0
	r7a.48xlarge	emr-5.36.1, emr-6.10.0

Kelas Instans	Tipe instans	Versi EMR Amazon yang didukung minimum (5.x, 6.x, 7.x)
	r7g.xlarge	emr-5.36.1, emr-6.10.0, emr-7.0.0
	r7g.2xlarge	emr-5.36.1, emr-6.10.0, emr-7.0.0
	r7g.4xlarge	emr-5.36.1, emr-6.10.0, emr-7.0.0
	r7g.8xlarge	emr-5.36.1, emr-6.10.0, emr-7.0.0
	r7g.12xlarge	emr-5.36.1, emr-6.10.0, emr-7.0.0
	r7g.16xlarge	emr-5.36.1, emr-6.10.0, emr-7.0.0
	r7gd.xlarge	emr-5.36.1, emr-6.10.0, emr-7.0.0
	r7gd.2xlarge	emr-5.36.1, emr-6.10.0, emr-7.0.0
	r7gd.4xlarge	emr-5.36.1, emr-6.10.0, emr-7.0.0
	r7gd.8xlarge	emr-5.36.1, emr-6.10.0, emr-7.0.0
	r7gd.12xlarge	emr-5.36.1, emr-6.10.0, emr-7.0.0
	r7gd.16xlarge	emr-5.36.1, emr-6.10.0, emr-7.0.0

Kelas Instans	Tipe instans	Versi EMR Amazon yang didukung minimum (5.x, 6.x, 7.x)
	r7i.xlarge	emr-5.36.1, emr-6.10.0
	r7i.2xlarge	emr-5.36.1, emr-6.10.0
	r7i.4xlarge	emr-5.36.1, emr-6.10.0
	r7i.8xlarge	emr-5.36.1, emr-6.10.0
	r7i.12xlarge	emr-5.36.1, emr-6.10.0
	r7i.16xlarge	emr-5.36.1, emr-6.10.0
	r7i.24xlarge	emr-5.36.1, emr-6.10.0
	r7i.48xlarge	emr-5.36.1, emr-6.10.0
	x1.16xlarge	emr-5.36.1, emr-6.9.0, emr-7.0.0
	x1.32xlarge	emr-5.36.1, emr-6.9.0, emr-7.0.0
	x1e.xlarge	emr-5.36.1, emr-6.10.0, emr-7.0.0
	x1e.2xlarge	emr-5.36.1, emr-6.10.0, emr-7.0.0
	x1e.4xlarge	emr-5.36.1, emr-6.10.0, emr-7.0.0
	x1e.8xlarge	emr-5.36.1, emr-6.10.0, emr-7.0.0
	x1e.16xlarge	emr-5.36.1, emr-6.10.0, emr-7.0.0

Kelas Instans	Tipe instans	Versi EMR Amazon yang didukung minimum (5.x, 6.x, 7.x)
	x1e.32xlarge	emr-5.36.1, emr-6.10.0, emr-7.0.0
	x2gd.xlarge	emr-5.36.1, emr-6.10.0, emr-7.0.0
	x2gd.2xlarge	emr-5.36.1, emr-6.10.0, emr-7.0.0
	x2gd.4xlarge	emr-5.36.1, emr-6.10.0, emr-7.0.0
	x2gd.8xlarge	emr-5.36.1, emr-6.10.0, emr-7.0.0
	x2gd.12xlarge	emr-5.36.1, emr-6.10.0, emr-7.0.0
	x2gd.16xlarge	emr-5.36.1, emr-6.10.0, emr-7.0.0
	x2idn.16xlarge	emr-5.36.1, emr-6.9.0
	x2idn.24xlarge	emr-5.36.1, emr-6.9.0
	x2idn.32xlarge	emr-5.36.1, emr-6.9.0
	x2iedn.xlarge	emr-5.36.1, emr-6.9.0
	x2iedn.2xlarge	emr-5.36.1, emr-6.9.0
	x2iedn.4xlarge	emr-5.36.1, emr-6.9.0
	x2iedn.8xlarge	emr-5.36.1, emr-6.9.0
	x2iedn.16xlarge	emr-5.36.1, emr-6.9.0

Kelas Instans	Tipe instans	Versi EMR Amazon yang didukung minimum (5.x, 6.x, 7.x)
	x2iedn.24xlarge	emr-5.36.1, emr-6.9.0
	x2iedn.32xlarge	emr-5.36.1, emr-6.9.0
	z1d.xlarge	emr-5.17.0, emr-6.0.0, emr-7.0.0
	z1d.2xlarge	emr-5.17.0, emr-6.0.0, emr-7.0.0
	z1d.3xlarge	emr-5.17.0, emr-6.0.0, emr-7.0.0
	z1d.6xlarge	emr-5.17.0, emr-6.0.0, emr-7.0.0
	z1d.12xlarge	emr-5.17.0, emr-6.0.0, emr-7.0.0
Penyimpanan Dioptimalkan	d3.xlarge	emr-5.33.0, emr-6.3.0, emr-7.0.0
	d3.2xlarge	emr-5.33.0, emr-6.3.0, emr-7.0.0
	d3.4xlarge	emr-5.33.0, emr-6.3.0, emr-7.0.0
	d3.8xlarge	emr-5.33.0, emr-6.3.0, emr-7.0.0
	h1.2xlarge	emr-5.17.0, emr-6.0.0, emr-7.0.0
	h1.4xlarge	emr-5.17.0, emr-6.0.0, emr-7.0.0

Kelas Instans	Tipe instans	Versi EMR Amazon yang didukung minimum (5.x, 6.x, 7.x)
	h1.8xlarge	emr-5.17.0, emr-6.0.0, emr-7.0.0
	h1.16xlarge	emr-5.17.0, emr-6.0.0, emr-7.0.0
	i3.xlarge	emr-5.9.0, emr-6.0.0, emr-7.0.0
	i3.2xlarge	emr-5.9.0, emr-6.0.0, emr-7.0.0
	i3.4xlarge	emr-5.9.0, emr-6.0.0, emr-7.0.0
	i3.8xlarge	emr-5.9.0, emr-6.0.0, emr-7.0.0
	i3.16xlarge	emr-5.9.0, emr-6.0.0, emr-7.0.0
	i3en.xlarge	emr-5.25.0, emr-6.0.0, emr-7.0.0
	i3en.2xlarge	emr-5.25.0, emr-6.0.0, emr-7.0.0
	i3en.3xlarge	emr-5.25.0, emr-6.0.0, emr-7.0.0
	i3en.6xlarge	emr-5.25.0, emr-6.0.0, emr-7.0.0
	i3en.12xlarge	emr-5.25.0, emr-6.0.0, emr-7.0.0

Kelas Instans	Tipe instans	Versi EMR Amazon yang didukung minimum (5.x, 6.x, 7.x)
	i3en.24xlarge	emr-5.25.0, emr-6.0.0, emr-7.0.0
	i4g.xlarge	emr-5.36.1, emr-6.10.0, emr-7.0.0
	i4g.2xlarge	emr-5.36.1, emr-6.10.0, emr-7.0.0
	i4g.4xlarge	emr-5.36.1, emr-6.10.0, emr-7.0.0
	i4g.8xlarge	emr-5.36.1, emr-6.10.0, emr-7.0.0
	i4g.16xlarge	emr-5.36.1, emr-6.10.0, emr-7.0.0
	i4i.xlarge	emr-5.36.1, emr-6.8.0
	i4i.2xlarge	emr-5.36.1, emr-6.8.0
	i4i.4xlarge	emr-5.36.1, emr-6.8.0
	i4i.8xlarge	emr-5.36.1, emr-6.8.0
	i4i.12xlarge	emr-5.36.1, emr-6.10.0
	i4i.16xlarge	emr-5.36.1, emr-6.8.0
	i4i.24xlarge	emr-5.36.1, emr-6.10.0
	i4i.32xlarge	emr-5.36.1, emr-6.8.0
	im4gn.xlarge	emr-5.35.0, emr-6.6.0, emr-7.0.0

Kelas Instans	Tipe instans	Versi EMR Amazon yang didukung minimum (5.x, 6.x, 7.x)
	im4gn.2xlarge	emr-5.35.0, emr-6.6.0, emr-7.0.0
	im4gn.4xlarge	emr-5.35.0, emr-6.6.0, emr-7.0.0
	im4gn.8xlarge	emr-5.35.0, emr-6.6.0, emr-7.0.0
	im4gn.16xlarge	emr-5.35.0, emr-6.6.0, emr-7.0.0
	is4gen.xlarge	emr-5.35.0, emr-6.6.0, emr-7.0.0
	is4gen.2xlarge	emr-5.35.0, emr-6.6.0, emr-7.0.0
	is4gen.4xlarge	emr-5.35.0, emr-6.6.0, emr-7.0.0
	is4gen.8xlarge	emr-5.35.0, emr-6.6.0, emr-7.0.0

US West (N. California) - us-west-1

Kelas Instans	Tipe instans	Versi EMR Amazon yang didukung minimum (5.x, 6.x, 7.x)
Tujuan Umum	m5.xlarge	emr-5.13.0, emr-6.0.0, emr-7.0.0

Kelas Instans	Tipe instans	Versi EMR Amazon yang didukung minimum (5.x, 6.x, 7.x)
	m5.2xlarge	emr-5.13.0, emr-6.0.0, emr-7.0.0
	m5.4xlarge	emr-5.13.0, emr-6.0.0, emr-7.0.0
	m5.8xlarge	emr-5.13.0, emr-6.0.0, emr-7.0.0
	m5.12xlarge	emr-5.13.0, emr-6.0.0, emr-7.0.0
	m5.16xlarge	emr-5.13.0, emr-6.0.0, emr-7.0.0
	m5.24xlarge	emr-5.13.0, emr-6.0.0, emr-7.0.0
	m5a.xlarge	emr-5.20.0, emr-6.0.0, emr-7.0.0
	m5a.2xlarge	emr-5.20.0, emr-6.0.0, emr-7.0.0
	m5a.4xlarge	emr-5.20.0, emr-6.0.0, emr-7.0.0
	m5a.8xlarge	emr-5.20.0, emr-6.0.0, emr-7.0.0
	m5a.12xlarge	emr-5.20.0, emr-6.0.0, emr-7.0.0
	m5a.16xlarge	emr-5.20.0, emr-6.0.0, emr-7.0.0

Kelas Instans	Tipe instans	Versi EMR Amazon yang didukung minimum (5.x, 6.x, 7.x)
	m5a.24xlarge	emr-5.20.0, emr-6.0.0, emr-7.0.0
	m5ad.xlarge	emr-5.20.0, emr-6.0.0, emr-7.0.0
	m5ad.2xlarge	emr-5.20.0, emr-6.0.0, emr-7.0.0
	m5ad.4xlarge	emr-5.20.0, emr-6.0.0, emr-7.0.0
	m5ad.8xlarge	emr-5.33.0, emr-6.3.0, emr-7.0.0
	m5ad.12xlarge	emr-5.20.0, emr-6.0.0, emr-7.0.0
	m5ad.16xlarge	emr-5.33.0, emr-6.3.0, emr-7.0.0
	m5ad.24xlarge	emr-5.20.0, emr-6.0.0, emr-7.0.0
	m5d.xlarge	emr-5.13.0, emr-6.0.0, emr-7.0.0
	m5d.2xlarge	emr-5.13.0, emr-6.0.0, emr-7.0.0
	m5d.4xlarge	emr-5.13.0, emr-6.0.0, emr-7.0.0
	m5d.8xlarge	emr-5.13.0, emr-6.0.0, emr-7.0.0

Kelas Instans	Tipe instans	Versi EMR Amazon yang didukung minimum (5.x, 6.x, 7.x)
	m5d.12xlarge	emr-5.13.0, emr-6.0.0, emr-7.0.0
	m5d.16xlarge	emr-5.13.0, emr-6.0.0, emr-7.0.0
	m5d.24xlarge	emr-5.13.0, emr-6.0.0, emr-7.0.0
	m5zn.xlarge	emr-5.33.0, emr-6.3.0, emr-7.0.0
	m5zn.2xlarge	emr-5.33.0, emr-6.3.0, emr-7.0.0
	m5zn.3xlarge	emr-5.33.0, emr-6.3.0, emr-7.0.0
	m5zn.6xlarge	emr-5.33.0, emr-6.3.0, emr-7.0.0
	m5zn.12xlarge	emr-5.33.0, emr-6.3.0, emr-7.0.0
	m6a.xlarge	emr-5.36.1, emr-6.8.0, emr-7.0.0
	m6a.2xlarge	emr-5.36.1, emr-6.8.0, emr-7.0.0
	m6a.4xlarge	emr-5.36.1, emr-6.8.0, emr-7.0.0
	m6a.8xlarge	emr-5.36.1, emr-6.8.0, emr-7.0.0

Kelas Instans	Tipe instans	Versi EMR Amazon yang didukung minimum (5.x, 6.x, 7.x)
	m6a.12xlarge	emr-5.36.1, emr-6.8.0, emr-7.0.0
	m6a.16xlarge	emr-5.36.1, emr-6.8.0, emr-7.0.0
	m6a.24xlarge	emr-5.36.1, emr-6.8.0, emr-7.0.0
	m6a.32xlarge	emr-5.36.1, emr-6.8.0, emr-7.0.0
	m6a.48xlarge	emr-5.36.1, emr-6.8.0, emr-7.0.0
	m6g.xlarge	emr-5.30.0, emr-6.1.0, emr-7.0.0
	m6g.2xlarge	emr-5.30.0, emr-6.1.0, emr-7.0.0
	m6g.4xlarge	emr-5.30.0, emr-6.1.0, emr-7.0.0
	m6g.8xlarge	emr-5.30.0, emr-6.1.0, emr-7.0.0
	m6g.12xlarge	emr-5.30.0, emr-6.1.0, emr-7.0.0
	m6g.16xlarge	emr-5.30.0, emr-6.1.0, emr-7.0.0
	m6gd.xlarge	emr-5.33.0, emr-6.3.0, emr-7.0.0

Kelas Instans	Tipe instans	Versi EMR Amazon yang didukung minimum (5.x, 6.x, 7.x)
	m6gd.2xlarge	emr-5.33.0, emr-6.3.0, emr-7.0.0
	m6gd.4xlarge	emr-5.33.0, emr-6.3.0, emr-7.0.0
	m6gd.8xlarge	emr-5.33.0, emr-6.3.0, emr-7.0.0
	m6gd.12xlarge	emr-5.33.0, emr-6.3.0, emr-7.0.0
	m6gd.16xlarge	emr-5.33.0, emr-6.3.0, emr-7.0.0
	m6i.xlarge	emr-5.35.0, emr-6.6.0
	m6i.2xlarge	emr-5.35.0, emr-6.6.0
	m6i.4xlarge	emr-5.35.0, emr-6.6.0
	m6i.8xlarge	emr-5.35.0, emr-6.6.0
	m6i.12xlarge	emr-5.35.0, emr-6.6.0
	m6i.16xlarge	emr-5.35.0, emr-6.6.0
	m6i.24xlarge	emr-5.35.0, emr-6.6.0
	m6i.32xlarge	emr-5.35.0, emr-6.6.0
	m6idn.xlarge	emr-5.36.1, emr-6.10.0
	m6idn.2xlarge	emr-5.36.1, emr-6.10.0
	m6idn.4xlarge	emr-5.36.1, emr-6.10.0

Kelas Instans	Tipe instans	Versi EMR Amazon yang didukung minimum (5.x, 6.x, 7.x)
	m6idn.8xlarge	emr-5.36.1, emr-6.10.0
	m6idn.12xlarge	emr-5.36.1, emr-6.10.0
	m6idn.16xlarge	emr-5.36.1, emr-6.10.0
	m6idn.24xlarge	emr-5.36.1, emr-6.10.0
	m6idn.32xlarge	emr-5.36.1, emr-6.10.0
	m6in.xlarge	emr-5.36.1, emr-6.10.0
	m6in.2xlarge	emr-5.36.1, emr-6.10.0
	m6in.4xlarge	emr-5.36.1, emr-6.10.0
	m6in.8xlarge	emr-5.36.1, emr-6.10.0
	m6in.12xlarge	emr-5.36.1, emr-6.10.0
	m6in.16xlarge	emr-5.36.1, emr-6.10.0
	m6in.24xlarge	emr-5.36.1, emr-6.10.0
	m6in.32xlarge	emr-5.36.1, emr-6.10.0
	m7g.xlarge	emr-5.36.1, emr-6.10.0, emr-7.0.0
	m7g.2xlarge	emr-5.36.1, emr-6.10.0, emr-7.0.0
	m7g.4xlarge	emr-5.36.1, emr-6.10.0, emr-7.0.0
	m7g.8xlarge	emr-5.36.1, emr-6.10.0, emr-7.0.0

Kelas Instans	Tipe instans	Versi EMR Amazon yang didukung minimum (5.x, 6.x, 7.x)
	m7g.12xlarge	emr-5.36.1, emr-6.10.0, emr-7.0.0
	m7g.16xlarge	emr-5.36.1, emr-6.10.0, emr-7.0.0
	m7gd.xlarge	emr-5.36.1, emr-6.10.0, emr-7.0.0
	m7gd.2xlarge	emr-5.36.1, emr-6.10.0, emr-7.0.0
	m7gd.4xlarge	emr-5.36.1, emr-6.10.0, emr-7.0.0
	m7gd.8xlarge	emr-5.36.1, emr-6.10.0, emr-7.0.0
	m7gd.12xlarge	emr-5.36.1, emr-6.10.0, emr-7.0.0
	m7gd.16xlarge	emr-5.36.1, emr-6.10.0, emr-7.0.0
Komputasi Dioptimalkan	c5.xlarge	emr-5.13.0, emr-6.0.0, emr-7.0.0
	c5.2xlarge	emr-5.13.0, emr-6.0.0, emr-7.0.0
	c5.4xlarge	emr-5.13.0, emr-6.0.0, emr-7.0.0
	c5.9xlarge	emr-5.13.0, emr-6.0.0, emr-7.0.0

Kelas Instans	Tipe instans	Versi EMR Amazon yang didukung minimum (5.x, 6.x, 7.x)
	c5.12xlarge	emr-5.13.0, emr-6.0.0, emr-7.0.0
	c5.18xlarge	emr-5.13.0, emr-6.0.0, emr-7.0.0
	c5.24xlarge	emr-5.13.0, emr-6.0.0, emr-7.0.0
	c5a.xlarge	emr-5.31.0, emr-6.1.0, emr-7.0.0
	c5a.2xlarge	emr-5.31.0, emr-6.1.0, emr-7.0.0
	c5a.4xlarge	emr-5.31.0, emr-6.1.0, emr-7.0.0
	c5a.8xlarge	emr-5.31.0, emr-6.1.0, emr-7.0.0
	c5a.12xlarge	emr-5.31.0, emr-6.1.0, emr-7.0.0
	c5a.16xlarge	emr-5.31.0, emr-6.1.0, emr-7.0.0
	c5a.24xlarge	emr-5.31.0, emr-6.1.0, emr-7.0.0
	c5d.xlarge	emr-5.13.0, emr-6.0.0, emr-7.0.0
	c5d.2xlarge	emr-5.13.0, emr-6.0.0, emr-7.0.0

Kelas Instans	Tipe instans	Versi EMR Amazon yang didukung minimum (5.x, 6.x, 7.x)
	c5d.4xlarge	emr-5.13.0, emr-6.0.0, emr-7.0.0
	c5d.9xlarge	emr-5.13.0, emr-6.0.0, emr-7.0.0
	c5d.12xlarge	emr-5.13.0, emr-6.0.0, emr-7.0.0
	c5d.18xlarge	emr-5.13.0, emr-6.0.0, emr-7.0.0
	c5d.24xlarge	emr-5.13.0, emr-6.0.0, emr-7.0.0
	c5n.xlarge	emr-5.20.0, emr-6.0.0, emr-7.0.0
	c5n.2xlarge	emr-5.20.0, emr-6.0.0, emr-7.0.0
	c5n.4xlarge	emr-5.20.0, emr-6.0.0, emr-7.0.0
	c5n.9xlarge	emr-5.20.0, emr-6.0.0, emr-7.0.0
	c5n.18xlarge	emr-5.20.0, emr-6.0.0, emr-7.0.0
	c6a.xlarge	emr-5.36.1, emr-6.8.0, emr-7.0.0
	c6a.2xlarge	emr-5.36.1, emr-6.8.0, emr-7.0.0

Kelas Instans	Tipe instans	Versi EMR Amazon yang didukung minimum (5.x, 6.x, 7.x)
	c6a.4xlarge	emr-5.36.1, emr-6.8.0, emr-7.0.0
	c6a.8xlarge	emr-5.36.1, emr-6.8.0, emr-7.0.0
	c6a.12xlarge	emr-5.36.1, emr-6.8.0, emr-7.0.0
	c6a.16xlarge	emr-5.36.1, emr-6.8.0, emr-7.0.0
	c6a.24xlarge	emr-5.36.1, emr-6.8.0, emr-7.0.0
	c6a.32xlarge	emr-5.36.1, emr-6.8.0, emr-7.0.0
	c6a.48xlarge	emr-5.36.1, emr-6.8.0, emr-7.0.0
	c6g.xlarge	emr-5.31.0, emr-6.1.0, emr-7.0.0
	c6g.2xlarge	emr-5.31.0, emr-6.1.0, emr-7.0.0
	c6g.4xlarge	emr-5.31.0, emr-6.1.0, emr-7.0.0
	c6g.8xlarge	emr-5.31.0, emr-6.1.0, emr-7.0.0
	c6g.12xlarge	emr-5.31.0, emr-6.1.0, emr-7.0.0

Kelas Instans	Tipe instans	Versi EMR Amazon yang didukung minimum (5.x, 6.x, 7.x)
	c6g.16xlarge	emr-5.31.0, emr-6.1.0, emr-7.0.0
	c6gd.xlarge	emr-5.33.0, emr-6.3.0, emr-7.0.0
	c6gd.2xlarge	emr-5.33.0, emr-6.3.0, emr-7.0.0
	c6gd.4xlarge	emr-5.33.0, emr-6.3.0, emr-7.0.0
	c6gd.8xlarge	emr-5.33.0, emr-6.3.0, emr-7.0.0
	c6gd.12xlarge	emr-5.33.0, emr-6.3.0, emr-7.0.0
	c6gd.16xlarge	emr-5.33.0, emr-6.3.0, emr-7.0.0
	c6gn.xlarge	emr-5.33.0, emr-6.3.0, emr-7.0.0
	c6gn.2xlarge	emr-5.33.0, emr-6.3.0, emr-7.0.0
	c6gn.4xlarge	emr-5.33.0, emr-6.3.0, emr-7.0.0
	c6gn.8xlarge	emr-5.33.0, emr-6.3.0, emr-7.0.0
	c6gn.12xlarge	emr-5.33.0, emr-6.3.0, emr-7.0.0

Kelas Instans	Tipe instans	Versi EMR Amazon yang didukung minimum (5.x, 6.x, 7.x)
	c6gn.16xlarge	emr-5.33.0, emr-6.3.0, emr-7.0.0
	c6i.xlarge	emr-5.35.0, emr-6.6.0
	c6i.2xlarge	emr-5.35.0, emr-6.6.0
	c6i.4xlarge	emr-5.35.0, emr-6.6.0
	c6i.8xlarge	emr-5.35.0, emr-6.6.0
	c6i.12xlarge	emr-5.35.0, emr-6.6.0
	c6i.16xlarge	emr-5.35.0, emr-6.6.0
	c6i.24xlarge	emr-5.35.0, emr-6.6.0
	c6i.32xlarge	emr-5.35.0, emr-6.6.0
	c6in.xlarge	emr-5.36.1, emr-6.10.0
	c6in.2xlarge	emr-5.36.1, emr-6.10.0
	c6in.4xlarge	emr-5.36.1, emr-6.10.0
	c6in.8xlarge	emr-5.36.1, emr-6.10.0
	c6in.12xlarge	emr-5.36.1, emr-6.10.0
	c6in.16xlarge	emr-5.36.1, emr-6.10.0
	c6in.24xlarge	emr-5.36.1, emr-6.10.0
	c6in.32xlarge	emr-5.36.1, emr-6.10.0
	c7g.xlarge	emr-5.36.1, emr-6.7.0, emr-7.0.0

Kelas Instans	Tipe instans	Versi EMR Amazon yang didukung minimum (5.x, 6.x, 7.x)
	c7g.2xlarge	emr-5.36.1, emr-6.7.0, emr-7.0.0
	c7g.4xlarge	emr-5.36.1, emr-6.7.0, emr-7.0.0
	c7g.8xlarge	emr-5.36.1, emr-6.7.0, emr-7.0.0
	c7g.12xlarge	emr-5.36.1, emr-6.7.0, emr-7.0.0
	c7g.16xlarge	emr-5.36.1, emr-6.7.0, emr-7.0.0
	c7gd.xlarge	emr-5.36.1, emr-6.10.0, emr-7.0.0
	c7gd.2xlarge	emr-5.36.1, emr-6.10.0, emr-7.0.0
	c7gd.4xlarge	emr-5.36.1, emr-6.10.0, emr-7.0.0
	c7gd.8xlarge	emr-5.36.1, emr-6.10.0, emr-7.0.0
	c7gd.12xlarge	emr-5.36.1, emr-6.10.0, emr-7.0.0
	c7gd.16xlarge	emr-5.36.1, emr-6.10.0, emr-7.0.0
Komputasi yang Dipercepat	g3.4xlarge	emr-5.18.0, emr-6.0.0, emr-7.0.0

Kelas Instans	Tipe instans	Versi EMR Amazon yang didukung minimum (5.x, 6.x, 7.x)
	g3.8xlarge	emr-5.18.0, emr-6.0.0, emr-7.0.0
	g3.16xlarge	emr-5.18.0, emr-6.0.0, emr-7.0.0
	g4dn.xlarge	emr-5.30.0, emr-6.0.0, emr-7.0.0
	g4dn.2xlarge	emr-5.30.0, emr-6.0.0, emr-7.0.0
	g4dn.4xlarge	emr-5.30.0, emr-6.0.0, emr-7.0.0
	g4dn.8xlarge	emr-5.30.0, emr-6.0.0, emr-7.0.0
	g4dn.12xlarge	emr-5.30.0, emr-6.0.0, emr-7.0.0
	g4dn.16xlarge	emr-5.30.0, emr-6.0.0, emr-7.0.0
Memori Dioptimalkan	r5.xlarge	emr-5.13.0, emr-6.0.0, emr-7.0.0
	r5.2xlarge	emr-5.13.0, emr-6.0.0, emr-7.0.0
	r5.4xlarge	emr-5.13.0, emr-6.0.0, emr-7.0.0
	r5.8xlarge	emr-5.13.0, emr-6.0.0, emr-7.0.0

Kelas Instans	Tipe instans	Versi EMR Amazon yang didukung minimum (5.x, 6.x, 7.x)
	r5.12xlarge	emr-5.13.0, emr-6.0.0, emr-7.0.0
	r5.16xlarge	emr-5.13.0, emr-6.0.0, emr-7.0.0
	r5.24xlarge	emr-5.13.0, emr-6.0.0, emr-7.0.0
	r5a.xlarge	emr-5.20.0, emr-6.0.0, emr-7.0.0
	r5a.2xlarge	emr-5.20.0, emr-6.0.0, emr-7.0.0
	r5a.4xlarge	emr-5.20.0, emr-6.0.0, emr-7.0.0
	r5a.8xlarge	emr-5.20.0, emr-6.0.0, emr-7.0.0
	r5a.12xlarge	emr-5.20.0, emr-6.0.0, emr-7.0.0
	r5a.16xlarge	emr-5.20.0, emr-6.0.0, emr-7.0.0
	r5a.24xlarge	emr-5.20.0, emr-6.0.0, emr-7.0.0
	r5ad.xlarge	emr-5.20.0, emr-6.0.0, emr-7.0.0
	r5ad.2xlarge	emr-5.20.0, emr-6.0.0, emr-7.0.0

Kelas Instans	Tipe instans	Versi EMR Amazon yang didukung minimum (5.x, 6.x, 7.x)
	r5ad.4xlarge	emr-5.20.0, emr-6.0.0, emr-7.0.0
	r5ad.8xlarge	emr-5.33.0, emr-6.3.0, emr-7.0.0
	r5ad.12xlarge	emr-5.20.0, emr-6.0.0, emr-7.0.0
	r5ad.16xlarge	emr-5.33.0, emr-6.3.0, emr-7.0.0
	r5ad.24xlarge	emr-5.20.0, emr-6.0.0, emr-7.0.0
	r5d.xlarge	emr-5.13.0, emr-6.0.0, emr-7.0.0
	r5d.2xlarge	emr-5.13.0, emr-6.0.0, emr-7.0.0
	r5d.4xlarge	emr-5.13.0, emr-6.0.0, emr-7.0.0
	r5d.8xlarge	emr-5.13.0, emr-6.0.0, emr-7.0.0
	r5d.12xlarge	emr-5.13.0, emr-6.0.0, emr-7.0.0
	r5d.16xlarge	emr-5.13.0, emr-6.0.0, emr-7.0.0
	r5d.24xlarge	emr-5.13.0, emr-6.0.0, emr-7.0.0

Kelas Instans	Tipe instans	Versi EMR Amazon yang didukung minimum (5.x, 6.x, 7.x)
	r5n.xlarge	emr-5.31.0, emr-6.1.0, emr-7.0.0
	r5n.2xlarge	emr-5.31.0, emr-6.1.0, emr-7.0.0
	r5n.4xlarge	emr-5.31.0, emr-6.1.0, emr-7.0.0
	r5n.8xlarge	emr-5.31.0, emr-6.1.0, emr-7.0.0
	r5n.12xlarge	emr-5.31.0, emr-6.1.0, emr-7.0.0
	r5n.16xlarge	emr-5.31.0, emr-6.1.0, emr-7.0.0
	r5n.24xlarge	emr-5.31.0, emr-6.1.0, emr-7.0.0
	r6a.xlarge	emr-5.36.1, emr-6.8.0, emr-7.0.0
	r6a.2xlarge	emr-5.36.1, emr-6.8.0, emr-7.0.0
	r6a.4xlarge	emr-5.36.1, emr-6.8.0, emr-7.0.0
	r6a.8xlarge	emr-5.36.1, emr-6.8.0, emr-7.0.0
	r6a.12xlarge	emr-5.36.1, emr-6.8.0, emr-7.0.0

Kelas Instans	Tipe instans	Versi EMR Amazon yang didukung minimum (5.x, 6.x, 7.x)
	r6a.16xlarge	emr-5.36.1, emr-6.8.0, emr-7.0.0
	r6a.24xlarge	emr-5.36.1, emr-6.8.0, emr-7.0.0
	r6a.32xlarge	emr-5.36.1, emr-6.8.0, emr-7.0.0
	r6a.48xlarge	emr-5.36.1, emr-6.8.0, emr-7.0.0
	r6g.xlarge	emr-5.31.0, emr-6.1.0, emr-7.0.0
	r6g.2xlarge	emr-5.31.0, emr-6.1.0, emr-7.0.0
	r6g.4xlarge	emr-5.31.0, emr-6.1.0, emr-7.0.0
	r6g.8xlarge	emr-5.31.0, emr-6.1.0, emr-7.0.0
	r6g.12xlarge	emr-5.31.0, emr-6.1.0, emr-7.0.0
	r6g.16xlarge	emr-5.31.0, emr-6.1.0, emr-7.0.0
	r6gd.xlarge	emr-5.33.0, emr-6.3.0, emr-7.0.0
	r6gd.2xlarge	emr-5.33.0, emr-6.3.0, emr-7.0.0

Kelas Instans	Tipe instans	Versi EMR Amazon yang didukung minimum (5.x, 6.x, 7.x)
	r6gd.4xlarge	emr-5.33.0, emr-6.3.0, emr-7.0.0
	r6gd.8xlarge	emr-5.33.0, emr-6.3.0, emr-7.0.0
	r6gd.12xlarge	emr-5.33.0, emr-6.3.0, emr-7.0.0
	r6gd.16xlarge	emr-5.33.0, emr-6.3.0, emr-7.0.0
	r6i.xlarge	emr-5.35.0, emr-6.6.0
	r6i.2xlarge	emr-5.35.0, emr-6.6.0
	r6i.4xlarge	emr-5.35.0, emr-6.6.0
	r6i.8xlarge	emr-5.35.0, emr-6.6.0
	r6i.12xlarge	emr-5.35.0, emr-6.6.0
	r6i.16xlarge	emr-5.35.0, emr-6.6.0
	r6i.24xlarge	emr-5.35.0, emr-6.6.0
	r6i.32xlarge	emr-5.35.0, emr-6.6.0
	r7g.xlarge	emr-5.36.1, emr-6.10.0, emr-7.0.0
	r7g.2xlarge	emr-5.36.1, emr-6.10.0, emr-7.0.0
	r7g.4xlarge	emr-5.36.1, emr-6.10.0, emr-7.0.0

Kelas Instans	Tipe instans	Versi EMR Amazon yang didukung minimum (5.x, 6.x, 7.x)
	r7g.8xlarge	emr-5.36.1, emr-6.10.0, emr-7.0.0
	r7g.12xlarge	emr-5.36.1, emr-6.10.0, emr-7.0.0
	r7g.16xlarge	emr-5.36.1, emr-6.10.0, emr-7.0.0
	r7gd.xlarge	emr-5.36.1, emr-6.10.0, emr-7.0.0
	r7gd.2xlarge	emr-5.36.1, emr-6.10.0, emr-7.0.0
	r7gd.4xlarge	emr-5.36.1, emr-6.10.0, emr-7.0.0
	r7gd.8xlarge	emr-5.36.1, emr-6.10.0, emr-7.0.0
	r7gd.12xlarge	emr-5.36.1, emr-6.10.0, emr-7.0.0
	r7gd.16xlarge	emr-5.36.1, emr-6.10.0, emr-7.0.0
	x2idn.16xlarge	emr-5.36.1, emr-6.9.0
	x2idn.24xlarge	emr-5.36.1, emr-6.9.0
	x2idn.32xlarge	emr-5.36.1, emr-6.9.0
	x2iedn.xlarge	emr-5.36.1, emr-6.9.0
	x2iedn.2xlarge	emr-5.36.1, emr-6.9.0

Kelas Instans	Tipe instans	Versi EMR Amazon yang didukung minimum (5.x, 6.x, 7.x)
	x2iedn.4xlarge	emr-5.36.1, emr-6.9.0
	x2iedn.8xlarge	emr-5.36.1, emr-6.9.0
	x2iedn.16xlarge	emr-5.36.1, emr-6.9.0
	x2iedn.24xlarge	emr-5.36.1, emr-6.9.0
	x2iedn.32xlarge	emr-5.36.1, emr-6.9.0
	z1d.xlarge	emr-5.17.0, emr-6.0.0, emr-7.0.0
	z1d.2xlarge	emr-5.17.0, emr-6.0.0, emr-7.0.0
	z1d.3xlarge	emr-5.17.0, emr-6.0.0, emr-7.0.0
	z1d.6xlarge	emr-5.17.0, emr-6.0.0, emr-7.0.0
	z1d.12xlarge	emr-5.17.0, emr-6.0.0, emr-7.0.0
Penyimpanan Dioptimalkan	i3.xlarge	emr-5.9.0, emr-6.0.0, emr-7.0.0
	i3.2xlarge	emr-5.9.0, emr-6.0.0, emr-7.0.0
	i3.4xlarge	emr-5.9.0, emr-6.0.0, emr-7.0.0
	i3.8xlarge	emr-5.9.0, emr-6.0.0, emr-7.0.0

Kelas Instans	Tipe instans	Versi EMR Amazon yang didukung minimum (5.x, 6.x, 7.x)
	i3.16xlarge	emr-5.9.0, emr-6.0.0, emr-7.0.0
	i3en.xlarge	emr-5.25.0, emr-6.0.0, emr-7.0.0
	i3en.2xlarge	emr-5.25.0, emr-6.0.0, emr-7.0.0
	i3en.3xlarge	emr-5.25.0, emr-6.0.0, emr-7.0.0
	i3en.6xlarge	emr-5.25.0, emr-6.0.0, emr-7.0.0
	i3en.12xlarge	emr-5.25.0, emr-6.0.0, emr-7.0.0
	i3en.24xlarge	emr-5.25.0, emr-6.0.0, emr-7.0.0
	i4i.xlarge	emr-5.36.1, emr-6.8.0
	i4i.2xlarge	emr-5.36.1, emr-6.8.0
	i4i.4xlarge	emr-5.36.1, emr-6.8.0
	i4i.8xlarge	emr-5.36.1, emr-6.8.0
	i4i.12xlarge	emr-5.36.1, emr-6.10.0
	i4i.16xlarge	emr-5.36.1, emr-6.8.0
	i4i.24xlarge	emr-5.36.1, emr-6.10.0
	i4i.32xlarge	emr-5.36.1, emr-6.8.0

US West (Oregon) - us-west-2

Kelas Instans	Tipe instans	Versi EMR Amazon yang didukung minimum (5.x, 6.x, 7.x)
Tujuan Umum	m5.xlarge	emr-5.13.0, emr-6.0.0, emr-7.0.0
	m5.2xlarge	emr-5.13.0, emr-6.0.0, emr-7.0.0
	m5.4xlarge	emr-5.13.0, emr-6.0.0, emr-7.0.0
	m5.8xlarge	emr-5.13.0, emr-6.0.0, emr-7.0.0
	m5.12xlarge	emr-5.13.0, emr-6.0.0, emr-7.0.0
	m5.16xlarge	emr-5.13.0, emr-6.0.0, emr-7.0.0
	m5.24xlarge	emr-5.13.0, emr-6.0.0, emr-7.0.0
	m5a.xlarge	emr-5.20.0, emr-6.0.0, emr-7.0.0
	m5a.2xlarge	emr-5.20.0, emr-6.0.0, emr-7.0.0
	m5a.4xlarge	emr-5.20.0, emr-6.0.0, emr-7.0.0
	m5a.8xlarge	emr-5.20.0, emr-6.0.0, emr-7.0.0

Kelas Instans	Tipe instans	Versi EMR Amazon yang didukung minimum (5.x, 6.x, 7.x)
	m5a.12xlarge	emr-5.20.0, emr-6.0.0, emr-7.0.0
	m5a.16xlarge	emr-5.20.0, emr-6.0.0, emr-7.0.0
	m5a.24xlarge	emr-5.20.0, emr-6.0.0, emr-7.0.0
	m5ad.xlarge	emr-5.20.0, emr-6.0.0, emr-7.0.0
	m5ad.2xlarge	emr-5.20.0, emr-6.0.0, emr-7.0.0
	m5ad.4xlarge	emr-5.20.0, emr-6.0.0, emr-7.0.0
	m5ad.8xlarge	emr-5.33.0, emr-6.3.0, emr-7.0.0
	m5ad.12xlarge	emr-5.20.0, emr-6.0.0, emr-7.0.0
	m5ad.16xlarge	emr-5.33.0, emr-6.3.0, emr-7.0.0
	m5ad.24xlarge	emr-5.20.0, emr-6.0.0, emr-7.0.0
	m5d.xlarge	emr-5.13.0, emr-6.0.0, emr-7.0.0
	m5d.2xlarge	emr-5.13.0, emr-6.0.0, emr-7.0.0

Kelas Instans	Tipe instans	Versi EMR Amazon yang didukung minimum (5.x, 6.x, 7.x)
	m5d.4xlarge	emr-5.13.0, emr-6.0.0, emr-7.0.0
	m5d.8xlarge	emr-5.13.0, emr-6.0.0, emr-7.0.0
	m5d.12xlarge	emr-5.13.0, emr-6.0.0, emr-7.0.0
	m5d.16xlarge	emr-5.13.0, emr-6.0.0, emr-7.0.0
	m5d.24xlarge	emr-5.13.0, emr-6.0.0, emr-7.0.0
	m5dn.xlarge	emr-5.31.0, emr-6.1.0, emr-7.0.0
	m5dn.2xlarge	emr-5.31.0, emr-6.1.0, emr-7.0.0
	m5dn.4xlarge	emr-5.31.0, emr-6.1.0, emr-7.0.0
	m5dn.8xlarge	emr-5.31.0, emr-6.1.0, emr-7.0.0
	m5dn.12xlarge	emr-5.31.0, emr-6.1.0, emr-7.0.0
	m5dn.16xlarge	emr-5.31.0, emr-6.1.0, emr-7.0.0
	m5dn.24xlarge	emr-5.31.0, emr-6.1.0, emr-7.0.0

Kelas Instans	Tipe instans	Versi EMR Amazon yang didukung minimum (5.x, 6.x, 7.x)
	m5n.xlarge	emr-5.31.0, emr-6.1.0, emr-7.0.0
	m5n.2xlarge	emr-5.31.0, emr-6.1.0, emr-7.0.0
	m5n.4xlarge	emr-5.31.0, emr-6.1.0, emr-7.0.0
	m5n.8xlarge	emr-5.31.0, emr-6.1.0, emr-7.0.0
	m5n.12xlarge	emr-5.31.0, emr-6.1.0, emr-7.0.0
	m5n.16xlarge	emr-5.31.0, emr-6.1.0, emr-7.0.0
	m5n.24xlarge	emr-5.31.0, emr-6.1.0, emr-7.0.0
	m5zn.xlarge	emr-5.33.0, emr-6.3.0, emr-7.0.0
	m5zn.2xlarge	emr-5.33.0, emr-6.3.0, emr-7.0.0
	m5zn.3xlarge	emr-5.33.0, emr-6.3.0, emr-7.0.0
	m5zn.6xlarge	emr-5.33.0, emr-6.3.0, emr-7.0.0
	m5zn.12xlarge	emr-5.33.0, emr-6.3.0, emr-7.0.0

Kelas Instans	Tipe instans	Versi EMR Amazon yang didukung minimum (5.x, 6.x, 7.x)
	m6a.xlarge	emr-5.36.1, emr-6.8.0, emr-7.0.0
	m6a.2xlarge	emr-5.36.1, emr-6.8.0, emr-7.0.0
	m6a.4xlarge	emr-5.36.1, emr-6.8.0, emr-7.0.0
	m6a.8xlarge	emr-5.36.1, emr-6.8.0, emr-7.0.0
	m6a.12xlarge	emr-5.36.1, emr-6.8.0, emr-7.0.0
	m6a.16xlarge	emr-5.36.1, emr-6.8.0, emr-7.0.0
	m6a.24xlarge	emr-5.36.1, emr-6.8.0, emr-7.0.0
	m6a.32xlarge	emr-5.36.1, emr-6.8.0, emr-7.0.0
	m6a.48xlarge	emr-5.36.1, emr-6.8.0, emr-7.0.0
	m6g.xlarge	emr-5.30.0, emr-6.1.0, emr-7.0.0
	m6g.2xlarge	emr-5.30.0, emr-6.1.0, emr-7.0.0
	m6g.4xlarge	emr-5.30.0, emr-6.1.0, emr-7.0.0

Kelas Instans	Tipe instans	Versi EMR Amazon yang didukung minimum (5.x, 6.x, 7.x)
	m6g.8xlarge	emr-5.30.0, emr-6.1.0, emr-7.0.0
	m6g.12xlarge	emr-5.30.0, emr-6.1.0, emr-7.0.0
	m6g.16xlarge	emr-5.30.0, emr-6.1.0, emr-7.0.0
	m6gd.xlarge	emr-5.33.0, emr-6.3.0, emr-7.0.0
	m6gd.2xlarge	emr-5.33.0, emr-6.3.0, emr-7.0.0
	m6gd.4xlarge	emr-5.33.0, emr-6.3.0, emr-7.0.0
	m6gd.8xlarge	emr-5.33.0, emr-6.3.0, emr-7.0.0
	m6gd.12xlarge	emr-5.33.0, emr-6.3.0, emr-7.0.0
	m6gd.16xlarge	emr-5.33.0, emr-6.3.0, emr-7.0.0
	m6i.xlarge	emr-5.35.0, emr-6.6.0
	m6i.2xlarge	emr-5.35.0, emr-6.6.0
	m6i.4xlarge	emr-5.35.0, emr-6.6.0
	m6i.8xlarge	emr-5.35.0, emr-6.6.0
	m6i.12xlarge	emr-5.35.0, emr-6.6.0

Kelas Instans	Tipe instans	Versi EMR Amazon yang didukung minimum (5.x, 6.x, 7.x)
	m6i.16xlarge	emr-5.35.0, emr-6.6.0
	m6i.24xlarge	emr-5.35.0, emr-6.6.0
	m6i.32xlarge	emr-5.35.0, emr-6.6.0
	m6id.xlarge	emr-5.36.1, emr-6.8.0
	m6id.2xlarge	emr-5.36.1, emr-6.8.0
	m6id.4xlarge	emr-5.36.1, emr-6.8.0
	m6id.8xlarge	emr-5.36.1, emr-6.8.0
	m6id.12xlarge	emr-5.36.1, emr-6.8.0
	m6id.16xlarge	emr-5.36.1, emr-6.8.0
	m6id.24xlarge	emr-5.36.1, emr-6.8.0
	m6id.32xlarge	emr-5.36.1, emr-6.8.0
	m6idn.xlarge	emr-5.36.1, emr-6.10.0
	m6idn.2xlarge	emr-5.36.1, emr-6.10.0
	m6idn.4xlarge	emr-5.36.1, emr-6.10.0
	m6idn.8xlarge	emr-5.36.1, emr-6.10.0
	m6idn.12xlarge	emr-5.36.1, emr-6.10.0
	m6idn.16xlarge	emr-5.36.1, emr-6.10.0
	m6idn.24xlarge	emr-5.36.1, emr-6.10.0
	m6idn.32xlarge	emr-5.36.1, emr-6.10.0

Kelas Instans	Tipe instans	Versi EMR Amazon yang didukung minimum (5.x, 6.x, 7.x)
	m6in.xlarge	emr-5.36.1, emr-6.10.0
	m6in.2xlarge	emr-5.36.1, emr-6.10.0
	m6in.4xlarge	emr-5.36.1, emr-6.10.0
	m6in.8xlarge	emr-5.36.1, emr-6.10.0
	m6in.12xlarge	emr-5.36.1, emr-6.10.0
	m6in.16xlarge	emr-5.36.1, emr-6.10.0
	m6in.24xlarge	emr-5.36.1, emr-6.10.0
	m6in.32xlarge	emr-5.36.1, emr-6.10.0
	m7a.xlarge	emr-5.36.1, emr-6.10.0
	m7a.2xlarge	emr-5.36.1, emr-6.10.0
	m7a.4xlarge	emr-5.36.1, emr-6.10.0
	m7a.8xlarge	emr-5.36.1, emr-6.10.0
	m7a.12xlarge	emr-5.36.1, emr-6.10.0
	m7a.16xlarge	emr-5.36.1, emr-6.10.0
	m7a.24xlarge	emr-5.36.1, emr-6.10.0
	m7a.32xlarge	emr-5.36.1, emr-6.10.0
	m7a.48xlarge	emr-5.36.1, emr-6.10.0
	m7g.xlarge	emr-5.36.1, emr-6.10.0, emr-7.0.0

Kelas Instans	Tipe instans	Versi EMR Amazon yang didukung minimum (5.x, 6.x, 7.x)
	m7g.2xlarge	emr-5.36.1, emr-6.10.0, emr-7.0.0
	m7g.4xlarge	emr-5.36.1, emr-6.10.0, emr-7.0.0
	m7g.8xlarge	emr-5.36.1, emr-6.10.0, emr-7.0.0
	m7g.12xlarge	emr-5.36.1, emr-6.10.0, emr-7.0.0
	m7g.16xlarge	emr-5.36.1, emr-6.10.0, emr-7.0.0
	m7gd.xlarge	emr-5.36.1, emr-6.10.0, emr-7.0.0
	m7gd.2xlarge	emr-5.36.1, emr-6.10.0, emr-7.0.0
	m7gd.4xlarge	emr-5.36.1, emr-6.10.0, emr-7.0.0
	m7gd.8xlarge	emr-5.36.1, emr-6.10.0, emr-7.0.0
	m7gd.12xlarge	emr-5.36.1, emr-6.10.0, emr-7.0.0
	m7gd.16xlarge	emr-5.36.1, emr-6.10.0, emr-7.0.0
	m7i.xlarge	emr-5.36.1, emr-6.10.0
	m7i.2xlarge	emr-5.36.1, emr-6.10.0

Kelas Instans	Tipe instans	Versi EMR Amazon yang didukung minimum (5.x, 6.x, 7.x)
	m7i.4xlarge	emr-5.36.1, emr-6.10.0
	m7i.8xlarge	emr-5.36.1, emr-6.10.0
	m7i.12xlarge	emr-5.36.1, emr-6.10.0
	m7i.16xlarge	emr-5.36.1, emr-6.10.0
	m7i.24xlarge	emr-5.36.1, emr-6.10.0
	m7i.48xlarge	emr-5.36.1, emr-6.10.0
	m7i-flex.xlarge	emr-5.36.1, emr-6.10.0
	m7i-flex.2xlarge	emr-5.36.1, emr-6.10.0
	m7i-flex.4xlarge	emr-5.36.1, emr-6.10.0
	m7i-flex.8xlarge	emr-5.36.1, emr-6.10.0
Komputasi Dioptimalkan	c5.xlarge	emr-5.13.0, emr-6.0.0, emr-7.0.0
	c5.2xlarge	emr-5.13.0, emr-6.0.0, emr-7.0.0
	c5.4xlarge	emr-5.13.0, emr-6.0.0, emr-7.0.0
	c5.9xlarge	emr-5.13.0, emr-6.0.0, emr-7.0.0
	c5.12xlarge	emr-5.13.0, emr-6.0.0, emr-7.0.0
	c5.18xlarge	emr-5.13.0, emr-6.0.0, emr-7.0.0

Kelas Instans	Tipe instans	Versi EMR Amazon yang didukung minimum (5.x, 6.x, 7.x)
	c5.24xlarge	emr-5.13.0, emr-6.0.0, emr-7.0.0
	c5a.xlarge	emr-5.31.0, emr-6.1.0, emr-7.0.0
	c5a.2xlarge	emr-5.31.0, emr-6.1.0, emr-7.0.0
	c5a.4xlarge	emr-5.31.0, emr-6.1.0, emr-7.0.0
	c5a.8xlarge	emr-5.31.0, emr-6.1.0, emr-7.0.0
	c5a.12xlarge	emr-5.31.0, emr-6.1.0, emr-7.0.0
	c5a.16xlarge	emr-5.31.0, emr-6.1.0, emr-7.0.0
	c5a.24xlarge	emr-5.31.0, emr-6.1.0, emr-7.0.0
	c5ad.xlarge	emr-5.33.0, emr-6.3.0, emr-7.0.0
	c5ad.2xlarge	emr-5.33.0, emr-6.3.0, emr-7.0.0
	c5ad.4xlarge	emr-5.33.0, emr-6.3.0, emr-7.0.0
	c5ad.8xlarge	emr-5.33.0, emr-6.3.0, emr-7.0.0

Kelas Instans	Tipe instans	Versi EMR Amazon yang didukung minimum (5.x, 6.x, 7.x)
	c5ad.12xlarge	emr-5.33.0, emr-6.3.0, emr-7.0.0
	c5ad.16xlarge	emr-5.33.0, emr-6.3.0, emr-7.0.0
	c5ad.24xlarge	emr-5.33.0, emr-6.3.0, emr-7.0.0
	c5d.xlarge	emr-5.13.0, emr-6.0.0, emr-7.0.0
	c5d.2xlarge	emr-5.13.0, emr-6.0.0, emr-7.0.0
	c5d.4xlarge	emr-5.13.0, emr-6.0.0, emr-7.0.0
	c5d.9xlarge	emr-5.13.0, emr-6.0.0, emr-7.0.0
	c5d.12xlarge	emr-5.13.0, emr-6.0.0, emr-7.0.0
	c5d.18xlarge	emr-5.13.0, emr-6.0.0, emr-7.0.0
	c5d.24xlarge	emr-5.13.0, emr-6.0.0, emr-7.0.0
	c5n.xlarge	emr-5.20.0, emr-6.0.0, emr-7.0.0
	c5n.2xlarge	emr-5.20.0, emr-6.0.0, emr-7.0.0

Kelas Instans	Tipe instans	Versi EMR Amazon yang didukung minimum (5.x, 6.x, 7.x)
	c5n.4xlarge	emr-5.20.0, emr-6.0.0, emr-7.0.0
	c5n.9xlarge	emr-5.20.0, emr-6.0.0, emr-7.0.0
	c5n.18xlarge	emr-5.20.0, emr-6.0.0, emr-7.0.0
	c6a.xlarge	emr-5.36.1, emr-6.8.0, emr-7.0.0
	c6a.2xlarge	emr-5.36.1, emr-6.8.0, emr-7.0.0
	c6a.4xlarge	emr-5.36.1, emr-6.8.0, emr-7.0.0
	c6a.8xlarge	emr-5.36.1, emr-6.8.0, emr-7.0.0
	c6a.12xlarge	emr-5.36.1, emr-6.8.0, emr-7.0.0
	c6a.16xlarge	emr-5.36.1, emr-6.8.0, emr-7.0.0
	c6a.24xlarge	emr-5.36.1, emr-6.8.0, emr-7.0.0
	c6a.32xlarge	emr-5.36.1, emr-6.8.0, emr-7.0.0
	c6a.48xlarge	emr-5.36.1, emr-6.8.0, emr-7.0.0

Kelas Instans	Tipe instans	Versi EMR Amazon yang didukung minimum (5.x, 6.x, 7.x)
	c6g.xlarge	emr-5.31.0, emr-6.1.0, emr-7.0.0
	c6g.2xlarge	emr-5.31.0, emr-6.1.0, emr-7.0.0
	c6g.4xlarge	emr-5.31.0, emr-6.1.0, emr-7.0.0
	c6g.8xlarge	emr-5.31.0, emr-6.1.0, emr-7.0.0
	c6g.12xlarge	emr-5.31.0, emr-6.1.0, emr-7.0.0
	c6g.16xlarge	emr-5.31.0, emr-6.1.0, emr-7.0.0
	c6gd.xlarge	emr-5.33.0, emr-6.3.0, emr-7.0.0
	c6gd.2xlarge	emr-5.33.0, emr-6.3.0, emr-7.0.0
	c6gd.4xlarge	emr-5.33.0, emr-6.3.0, emr-7.0.0
	c6gd.8xlarge	emr-5.33.0, emr-6.3.0, emr-7.0.0
	c6gd.12xlarge	emr-5.33.0, emr-6.3.0, emr-7.0.0
	c6gd.16xlarge	emr-5.33.0, emr-6.3.0, emr-7.0.0

Kelas Instans	Tipe instans	Versi EMR Amazon yang didukung minimum (5.x, 6.x, 7.x)
	c6gn.xlarge	emr-5.33.0, emr-6.3.0, emr-7.0.0
	c6gn.2xlarge	emr-5.33.0, emr-6.3.0, emr-7.0.0
	c6gn.4xlarge	emr-5.33.0, emr-6.3.0, emr-7.0.0
	c6gn.8xlarge	emr-5.33.0, emr-6.3.0, emr-7.0.0
	c6gn.12xlarge	emr-5.33.0, emr-6.3.0, emr-7.0.0
	c6gn.16xlarge	emr-5.33.0, emr-6.3.0, emr-7.0.0
	c6i.xlarge	emr-5.35.0, emr-6.6.0
	c6i.2xlarge	emr-5.35.0, emr-6.6.0
	c6i.4xlarge	emr-5.35.0, emr-6.6.0
	c6i.8xlarge	emr-5.35.0, emr-6.6.0
	c6i.12xlarge	emr-5.35.0, emr-6.6.0
	c6i.16xlarge	emr-5.35.0, emr-6.6.0
	c6i.24xlarge	emr-5.35.0, emr-6.6.0
	c6i.32xlarge	emr-5.35.0, emr-6.6.0
	c6id.xlarge	emr-5.36.1, emr-6.8.0
	c6id.2xlarge	emr-5.36.1, emr-6.8.0

Kelas Instans	Tipe instans	Versi EMR Amazon yang didukung minimum (5.x, 6.x, 7.x)
	c6id.4xlarge	emr-5.36.1, emr-6.8.0
	c6id.8xlarge	emr-5.36.1, emr-6.8.0
	c6id.12xlarge	emr-5.36.1, emr-6.8.0
	c6id.16xlarge	emr-5.36.1, emr-6.8.0
	c6id.24xlarge	emr-5.36.1, emr-6.8.0
	c6id.32xlarge	emr-5.36.1, emr-6.8.0
	c6in.xlarge	emr-5.36.1, emr-6.10.0
	c6in.2xlarge	emr-5.36.1, emr-6.10.0
	c6in.4xlarge	emr-5.36.1, emr-6.10.0
	c6in.8xlarge	emr-5.36.1, emr-6.10.0
	c6in.12xlarge	emr-5.36.1, emr-6.10.0
	c6in.16xlarge	emr-5.36.1, emr-6.10.0
	c6in.24xlarge	emr-5.36.1, emr-6.10.0
	c6in.32xlarge	emr-5.36.1, emr-6.10.0
	c7a.xlarge	emr-5.36.1, emr-6.10.0
	c7a.2xlarge	emr-5.36.1, emr-6.10.0
	c7a.4xlarge	emr-5.36.1, emr-6.10.0
	c7a.8xlarge	emr-5.36.1, emr-6.10.0
	c7a.12xlarge	emr-5.36.1, emr-6.10.0

Kelas Instans	Tipe instans	Versi EMR Amazon yang didukung minimum (5.x, 6.x, 7.x)
	c7a.16xlarge	emr-5.36.1, emr-6.10.0
	c7a.24xlarge	emr-5.36.1, emr-6.10.0
	c7a.32xlarge	emr-5.36.1, emr-6.10.0
	c7a.48xlarge	emr-5.36.1, emr-6.10.0
	c7g.xlarge	emr-5.36.1, emr-6.7.0, emr-7.0.0
	c7g.2xlarge	emr-5.36.1, emr-6.7.0, emr-7.0.0
	c7g.4xlarge	emr-5.36.1, emr-6.7.0, emr-7.0.0
	c7g.8xlarge	emr-5.36.1, emr-6.7.0, emr-7.0.0
	c7g.12xlarge	emr-5.36.1, emr-6.7.0, emr-7.0.0
	c7g.16xlarge	emr-5.36.1, emr-6.7.0, emr-7.0.0
	c7gd.xlarge	emr-5.36.1, emr-6.10.0, emr-7.0.0
	c7gd.2xlarge	emr-5.36.1, emr-6.10.0, emr-7.0.0
	c7gd.4xlarge	emr-5.36.1, emr-6.10.0, emr-7.0.0

Kelas Instans	Tipe instans	Versi EMR Amazon yang didukung minimum (5.x, 6.x, 7.x)
	c7gd.8xlarge	emr-5.36.1, emr-6.10.0, emr-7.0.0
	c7gd.12xlarge	emr-5.36.1, emr-6.10.0, emr-7.0.0
	c7gd.16xlarge	emr-5.36.1, emr-6.10.0, emr-7.0.0
	c7gn.xlarge	emr-5.36.1, emr-6.10.0, emr-7.0.0
	c7gn.2xlarge	emr-5.36.1, emr-6.10.0, emr-7.0.0
	c7gn.4xlarge	emr-5.36.1, emr-6.10.0, emr-7.0.0
	c7gn.8xlarge	emr-5.36.1, emr-6.10.0, emr-7.0.0
	c7gn.12xlarge	emr-5.36.1, emr-6.10.0, emr-7.0.0
	c7gn.16xlarge	emr-5.36.1, emr-6.10.0, emr-7.0.0
	c7i.xlarge	emr-5.36.1, emr-6.10.0
	c7i.2xlarge	emr-5.36.1, emr-6.10.0
	c7i.4xlarge	emr-5.36.1, emr-6.10.0
	c7i.8xlarge	emr-5.36.1, emr-6.10.0
	c7i.12xlarge	emr-5.36.1, emr-6.10.0

Kelas Instans	Tipe instans	Versi EMR Amazon yang didukung minimum (5.x, 6.x, 7.x)
	c7i.16xlarge	emr-5.36.1, emr-6.10.0
	c7i.24xlarge	emr-5.36.1, emr-6.10.0
	c7i.48xlarge	emr-5.36.1, emr-6.10.0
Komputasi yang Dipercepat	g3.4xlarge	emr-5.18.0, emr-6.0.0, emr-7.0.0
	g3.8xlarge	emr-5.18.0, emr-6.0.0, emr-7.0.0
	g3.16xlarge	emr-5.18.0, emr-6.0.0, emr-7.0.0
	g3s.xlarge	emr-5.19.0, emr-6.0.0, emr-7.0.0
	g4dn.xlarge	emr-5.30.0, emr-6.0.0, emr-7.0.0
	g4dn.2xlarge	emr-5.30.0, emr-6.0.0, emr-7.0.0
	g4dn.4xlarge	emr-5.30.0, emr-6.0.0, emr-7.0.0
	g4dn.8xlarge	emr-5.30.0, emr-6.0.0, emr-7.0.0
	g4dn.12xlarge	emr-5.30.0, emr-6.0.0, emr-7.0.0
	g4dn.16xlarge	emr-5.30.0, emr-6.0.0, emr-7.0.0

Kelas Instans	Tipe instans	Versi EMR Amazon yang didukung minimum (5.x, 6.x, 7.x)
	g5.xlarge	emr-5.36.1, emr-6.9.0, emr-7.0.0
	g5.2xlarge	emr-5.36.1, emr-6.9.0, emr-7.0.0
	g5.4xlarge	emr-5.36.1, emr-6.9.0, emr-7.0.0
	g5.8xlarge	emr-5.36.1, emr-6.9.0, emr-7.0.0
	g5.12xlarge	emr-5.36.1, emr-6.9.0, emr-7.0.0
	g5.16xlarge	emr-5.36.1, emr-6.9.0, emr-7.0.0
	g5.24xlarge	emr-5.36.1, emr-6.9.0, emr-7.0.0
	g5.48xlarge	emr-5.36.1, emr-6.9.0, emr-7.0.0
	p2.xlarge	emr-5.10.0, emr-6.0.0, emr-7.0.0
	p2.8xlarge	emr-5.10.0, emr-6.0.0, emr-7.0.0
	p2.16xlarge	emr-5.10.0, emr-6.0.0, emr-7.0.0
	p3.2xlarge	emr-5.10.0, emr-6.0.0, emr-7.0.0

Kelas Instans	Tipe instans	Versi EMR Amazon yang didukung minimum (5.x, 6.x, 7.x)
	p3.8xlarge	emr-5.10.0, emr-6.0.0, emr-7.0.0
	p3.16xlarge	emr-5.10.0, emr-6.0.0, emr-7.0.0
	p5.48xlarge	emr-6.14.0, emr-7.0.0
Memori Dioptimalkan	r5.xlarge	emr-5.13.0, emr-6.0.0, emr-7.0.0
	r5.2xlarge	emr-5.13.0, emr-6.0.0, emr-7.0.0
	r5.4xlarge	emr-5.13.0, emr-6.0.0, emr-7.0.0
	r5.8xlarge	emr-5.13.0, emr-6.0.0, emr-7.0.0
	r5.12xlarge	emr-5.13.0, emr-6.0.0, emr-7.0.0
	r5.16xlarge	emr-5.13.0, emr-6.0.0, emr-7.0.0
	r5.24xlarge	emr-5.13.0, emr-6.0.0, emr-7.0.0
	r5a.xlarge	emr-5.20.0, emr-6.0.0, emr-7.0.0
	r5a.2xlarge	emr-5.20.0, emr-6.0.0, emr-7.0.0

Kelas Instans	Tipe instans	Versi EMR Amazon yang didukung minimum (5.x, 6.x, 7.x)
	r5a.4xlarge	emr-5.20.0, emr-6.0.0, emr-7.0.0
	r5a.8xlarge	emr-5.20.0, emr-6.0.0, emr-7.0.0
	r5a.12xlarge	emr-5.20.0, emr-6.0.0, emr-7.0.0
	r5a.16xlarge	emr-5.20.0, emr-6.0.0, emr-7.0.0
	r5a.24xlarge	emr-5.20.0, emr-6.0.0, emr-7.0.0
	r5ad.xlarge	emr-5.20.0, emr-6.0.0, emr-7.0.0
	r5ad.2xlarge	emr-5.20.0, emr-6.0.0, emr-7.0.0
	r5ad.4xlarge	emr-5.20.0, emr-6.0.0, emr-7.0.0
	r5ad.8xlarge	emr-5.33.0, emr-6.3.0, emr-7.0.0
	r5ad.12xlarge	emr-5.20.0, emr-6.0.0, emr-7.0.0
	r5ad.16xlarge	emr-5.33.0, emr-6.3.0, emr-7.0.0
	r5ad.24xlarge	emr-5.20.0, emr-6.0.0, emr-7.0.0

Kelas Instans	Tipe instans	Versi EMR Amazon yang didukung minimum (5.x, 6.x, 7.x)
	r5b.xlarge	emr-5.33.0, emr-6.3.0, emr-7.0.0
	r5b.2xlarge	emr-5.33.0, emr-6.3.0, emr-7.0.0
	r5b.4xlarge	emr-5.33.0, emr-6.3.0, emr-7.0.0
	r5b.8xlarge	emr-5.33.0, emr-6.3.0, emr-7.0.0
	r5b.12xlarge	emr-5.33.0, emr-6.3.0, emr-7.0.0
	r5b.16xlarge	emr-5.33.0, emr-6.3.0, emr-7.0.0
	r5b.24xlarge	emr-5.33.0, emr-6.3.0, emr-7.0.0
	r5d.xlarge	emr-5.13.0, emr-6.0.0, emr-7.0.0
	r5d.2xlarge	emr-5.13.0, emr-6.0.0, emr-7.0.0
	r5d.4xlarge	emr-5.13.0, emr-6.0.0, emr-7.0.0
	r5d.8xlarge	emr-5.13.0, emr-6.0.0, emr-7.0.0
	r5d.12xlarge	emr-5.13.0, emr-6.0.0, emr-7.0.0

Kelas Instans	Tipe instans	Versi EMR Amazon yang didukung minimum (5.x, 6.x, 7.x)
	r5d.16xlarge	emr-5.13.0, emr-6.0.0, emr-7.0.0
	r5d.24xlarge	emr-5.13.0, emr-6.0.0, emr-7.0.0
	r5dn.xlarge	emr-5.31.0, emr-6.1.0, emr-7.0.0
	r5dn.2xlarge	emr-5.31.0, emr-6.1.0, emr-7.0.0
	r5dn.4xlarge	emr-5.31.0, emr-6.1.0, emr-7.0.0
	r5dn.8xlarge	emr-5.31.0, emr-6.1.0, emr-7.0.0
	r5dn.12xlarge	emr-5.31.0, emr-6.1.0, emr-7.0.0
	r5dn.16xlarge	emr-5.31.0, emr-6.1.0, emr-7.0.0
	r5dn.24xlarge	emr-5.31.0, emr-6.1.0, emr-7.0.0
	r5n.xlarge	emr-5.31.0, emr-6.1.0, emr-7.0.0
	r5n.2xlarge	emr-5.31.0, emr-6.1.0, emr-7.0.0
	r5n.4xlarge	emr-5.31.0, emr-6.1.0, emr-7.0.0

Kelas Instans	Tipe instans	Versi EMR Amazon yang didukung minimum (5.x, 6.x, 7.x)
	r5n.8xlarge	emr-5.31.0, emr-6.1.0, emr-7.0.0
	r5n.12xlarge	emr-5.31.0, emr-6.1.0, emr-7.0.0
	r5n.16xlarge	emr-5.31.0, emr-6.1.0, emr-7.0.0
	r5n.24xlarge	emr-5.31.0, emr-6.1.0, emr-7.0.0
	r6a.xlarge	emr-5.36.1, emr-6.8.0, emr-7.0.0
	r6a.2xlarge	emr-5.36.1, emr-6.8.0, emr-7.0.0
	r6a.4xlarge	emr-5.36.1, emr-6.8.0, emr-7.0.0
	r6a.8xlarge	emr-5.36.1, emr-6.8.0, emr-7.0.0
	r6a.12xlarge	emr-5.36.1, emr-6.8.0, emr-7.0.0
	r6a.16xlarge	emr-5.36.1, emr-6.8.0, emr-7.0.0
	r6a.24xlarge	emr-5.36.1, emr-6.8.0, emr-7.0.0
	r6a.32xlarge	emr-5.36.1, emr-6.8.0, emr-7.0.0

Kelas Instans	Tipe instans	Versi EMR Amazon yang didukung minimum (5.x, 6.x, 7.x)
	r6a.48xlarge	emr-5.36.1, emr-6.8.0, emr-7.0.0
	r6g.xlarge	emr-5.31.0, emr-6.1.0, emr-7.0.0
	r6g.2xlarge	emr-5.31.0, emr-6.1.0, emr-7.0.0
	r6g.4xlarge	emr-5.31.0, emr-6.1.0, emr-7.0.0
	r6g.8xlarge	emr-5.31.0, emr-6.1.0, emr-7.0.0
	r6g.12xlarge	emr-5.31.0, emr-6.1.0, emr-7.0.0
	r6g.16xlarge	emr-5.31.0, emr-6.1.0, emr-7.0.0
	r6gd.xlarge	emr-5.33.0, emr-6.3.0, emr-7.0.0
	r6gd.2xlarge	emr-5.33.0, emr-6.3.0, emr-7.0.0
	r6gd.4xlarge	emr-5.33.0, emr-6.3.0, emr-7.0.0
	r6gd.8xlarge	emr-5.33.0, emr-6.3.0, emr-7.0.0
	r6gd.12xlarge	emr-5.33.0, emr-6.3.0, emr-7.0.0

Kelas Instans	Tipe instans	Versi EMR Amazon yang didukung minimum (5.x, 6.x, 7.x)
	r6gd.16xlarge	emr-5.33.0, emr-6.3.0, emr-7.0.0
	r6i.xlarge	emr-5.35.0, emr-6.6.0
	r6i.2xlarge	emr-5.35.0, emr-6.6.0
	r6i.4xlarge	emr-5.35.0, emr-6.6.0
	r6i.8xlarge	emr-5.35.0, emr-6.6.0
	r6i.12xlarge	emr-5.35.0, emr-6.6.0
	r6i.16xlarge	emr-5.35.0, emr-6.6.0
	r6i.24xlarge	emr-5.35.0, emr-6.6.0
	r6i.32xlarge	emr-5.35.0, emr-6.6.0
	r6id.xlarge	emr-5.36.1, emr-6.8.0
	r6id.2xlarge	emr-5.36.1, emr-6.8.0
	r6id.4xlarge	emr-5.36.1, emr-6.8.0
	r6id.8xlarge	emr-5.36.1, emr-6.8.0
	r6id.12xlarge	emr-5.36.1, emr-6.8.0
	r6id.16xlarge	emr-5.36.1, emr-6.8.0
	r6id.24xlarge	emr-5.36.1, emr-6.8.0
	r6id.32xlarge	emr-5.36.1, emr-6.8.0
	r6idn.xlarge	emr-5.36.1, emr-6.10.0
	r6idn.2xlarge	emr-5.36.1, emr-6.10.0

Kelas Instans	Tipe instans	Versi EMR Amazon yang didukung minimum (5.x, 6.x, 7.x)
	r6idn.4xlarge	emr-5.36.1, emr-6.10.0
	r6idn.8xlarge	emr-5.36.1, emr-6.10.0
	r6idn.12xlarge	emr-5.36.1, emr-6.10.0
	r6idn.16xlarge	emr-5.36.1, emr-6.10.0
	r6idn.24xlarge	emr-5.36.1, emr-6.10.0
	r6idn.32xlarge	emr-5.36.1, emr-6.10.0
	r6in.xlarge	emr-5.36.1, emr-6.10.0
	r6in.2xlarge	emr-5.36.1, emr-6.10.0
	r6in.4xlarge	emr-5.36.1, emr-6.10.0
	r6in.8xlarge	emr-5.36.1, emr-6.10.0
	r6in.12xlarge	emr-5.36.1, emr-6.10.0
	r6in.16xlarge	emr-5.36.1, emr-6.10.0
	r6in.24xlarge	emr-5.36.1, emr-6.10.0
	r6in.32xlarge	emr-5.36.1, emr-6.10.0
	r7a.xlarge	emr-5.36.1, emr-6.10.0
	r7a.2xlarge	emr-5.36.1, emr-6.10.0
	r7a.4xlarge	emr-5.36.1, emr-6.10.0
	r7a.8xlarge	emr-5.36.1, emr-6.10.0
	r7a.12xlarge	emr-5.36.1, emr-6.10.0

Kelas Instans	Tipe instans	Versi EMR Amazon yang didukung minimum (5.x, 6.x, 7.x)
	r7a.16xlarge	emr-5.36.1, emr-6.10.0
	r7a.24xlarge	emr-5.36.1, emr-6.10.0
	r7a.32xlarge	emr-5.36.1, emr-6.10.0
	r7a.48xlarge	emr-5.36.1, emr-6.10.0
	r7g.xlarge	emr-5.36.1, emr-6.10.0, emr-7.0.0
	r7g.2xlarge	emr-5.36.1, emr-6.10.0, emr-7.0.0
	r7g.4xlarge	emr-5.36.1, emr-6.10.0, emr-7.0.0
	r7g.8xlarge	emr-5.36.1, emr-6.10.0, emr-7.0.0
	r7g.12xlarge	emr-5.36.1, emr-6.10.0, emr-7.0.0
	r7g.16xlarge	emr-5.36.1, emr-6.10.0, emr-7.0.0
	r7gd.xlarge	emr-5.36.1, emr-6.10.0, emr-7.0.0
	r7gd.2xlarge	emr-5.36.1, emr-6.10.0, emr-7.0.0
	r7gd.4xlarge	emr-5.36.1, emr-6.10.0, emr-7.0.0

Kelas Instans	Tipe instans	Versi EMR Amazon yang didukung minimum (5.x, 6.x, 7.x)
	r7gd.8xlarge	emr-5.36.1, emr-6.10.0, emr-7.0.0
	r7gd.12xlarge	emr-5.36.1, emr-6.10.0, emr-7.0.0
	r7gd.16xlarge	emr-5.36.1, emr-6.10.0, emr-7.0.0
	r7i.xlarge	emr-5.36.1, emr-6.10.0
	r7i.2xlarge	emr-5.36.1, emr-6.10.0
	r7i.4xlarge	emr-5.36.1, emr-6.10.0
	r7i.8xlarge	emr-5.36.1, emr-6.10.0
	r7i.12xlarge	emr-5.36.1, emr-6.10.0
	r7i.16xlarge	emr-5.36.1, emr-6.10.0
	r7i.24xlarge	emr-5.36.1, emr-6.10.0
	r7i.48xlarge	emr-5.36.1, emr-6.10.0
	r7iz.xlarge	emr-5.36.1, emr-6.10.0
	r7iz.2xlarge	emr-5.36.1, emr-6.10.0
	r7iz.4xlarge	emr-5.36.1, emr-6.10.0
	r7iz.8xlarge	emr-5.36.1, emr-6.10.0
	r7iz.12xlarge	emr-5.36.1, emr-6.10.0
	r7iz.16xlarge	emr-5.36.1, emr-6.10.0

Kelas Instans	Tipe instans	Versi EMR Amazon yang didukung minimum (5.x, 6.x, 7.x)
	r7iz.32xlarge	emr-5.36.1, emr-6.10.0
	x1.16xlarge	emr-5.36.1, emr-6.9.0, emr-7.0.0
	x1.32xlarge	emr-5.36.1, emr-6.9.0, emr-7.0.0
	x1e.xlarge	emr-5.36.1, emr-6.10.0, emr-7.0.0
	x1e.2xlarge	emr-5.36.1, emr-6.10.0, emr-7.0.0
	x1e.4xlarge	emr-5.36.1, emr-6.10.0, emr-7.0.0
	x1e.8xlarge	emr-5.36.1, emr-6.10.0, emr-7.0.0
	x1e.16xlarge	emr-5.36.1, emr-6.10.0, emr-7.0.0
	x1e.32xlarge	emr-5.36.1, emr-6.10.0, emr-7.0.0
	x2gd.xlarge	emr-5.36.1, emr-6.10.0, emr-7.0.0
	x2gd.2xlarge	emr-5.36.1, emr-6.10.0, emr-7.0.0
	x2gd.4xlarge	emr-5.36.1, emr-6.10.0, emr-7.0.0

Kelas Instans	Tipe instans	Versi EMR Amazon yang didukung minimum (5.x, 6.x, 7.x)
	x2gd.8xlarge	emr-5.36.1, emr-6.10.0, emr-7.0.0
	x2gd.12xlarge	emr-5.36.1, emr-6.10.0, emr-7.0.0
	x2gd.16xlarge	emr-5.36.1, emr-6.10.0, emr-7.0.0
	x2idn.16xlarge	emr-5.36.1, emr-6.9.0
	x2idn.24xlarge	emr-5.36.1, emr-6.9.0
	x2idn.32xlarge	emr-5.36.1, emr-6.9.0
	x2iedn.xlarge	emr-5.36.1, emr-6.9.0
	x2iedn.2xlarge	emr-5.36.1, emr-6.9.0
	x2iedn.4xlarge	emr-5.36.1, emr-6.9.0
	x2iedn.8xlarge	emr-5.36.1, emr-6.9.0
	x2iedn.16xlarge	emr-5.36.1, emr-6.9.0
	x2iedn.24xlarge	emr-5.36.1, emr-6.9.0
	x2iedn.32xlarge	emr-5.36.1, emr-6.9.0
	z1d.xlarge	emr-5.17.0, emr-6.0.0, emr-7.0.0
	z1d.2xlarge	emr-5.17.0, emr-6.0.0, emr-7.0.0
	z1d.3xlarge	emr-5.17.0, emr-6.0.0, emr-7.0.0

Kelas Instans	Tipe instans	Versi EMR Amazon yang didukung minimum (5.x, 6.x, 7.x)
	z1d.6xlarge	emr-5.17.0, emr-6.0.0, emr-7.0.0
	z1d.12xlarge	emr-5.17.0, emr-6.0.0, emr-7.0.0
Penyimpanan Dioptimalkan	d3.xlarge	emr-5.33.0, emr-6.3.0, emr-7.0.0
	d3.2xlarge	emr-5.33.0, emr-6.3.0, emr-7.0.0
	d3.4xlarge	emr-5.33.0, emr-6.3.0, emr-7.0.0
	d3.8xlarge	emr-5.33.0, emr-6.3.0, emr-7.0.0
	d3en.xlarge	emr-5.33.0, emr-6.3.0, emr-7.0.0
	d3en.2xlarge	emr-5.33.0, emr-6.3.0, emr-7.0.0
	d3en.4xlarge	emr-5.33.0, emr-6.3.0, emr-7.0.0
	d3en.6xlarge	emr-5.33.0, emr-6.3.0, emr-7.0.0
	d3en.8xlarge	emr-5.33.0, emr-6.3.0, emr-7.0.0
	d3en.12xlarge	emr-5.33.0, emr-6.3.0, emr-7.0.0

Kelas Instans	Tipe instans	Versi EMR Amazon yang didukung minimum (5.x, 6.x, 7.x)
	h1.2xlarge	emr-5.17.0, emr-6.0.0, emr-7.0.0
	h1.4xlarge	emr-5.17.0, emr-6.0.0, emr-7.0.0
	h1.8xlarge	emr-5.17.0, emr-6.0.0, emr-7.0.0
	h1.16xlarge	emr-5.17.0, emr-6.0.0, emr-7.0.0
	i3.xlarge	emr-5.9.0, emr-6.0.0, emr-7.0.0
	i3.2xlarge	emr-5.9.0, emr-6.0.0, emr-7.0.0
	i3.4xlarge	emr-5.9.0, emr-6.0.0, emr-7.0.0
	i3.8xlarge	emr-5.9.0, emr-6.0.0, emr-7.0.0
	i3.16xlarge	emr-5.9.0, emr-6.0.0, emr-7.0.0
	i3en.xlarge	emr-5.25.0, emr-6.0.0, emr-7.0.0
	i3en.2xlarge	emr-5.25.0, emr-6.0.0, emr-7.0.0
	i3en.3xlarge	emr-5.25.0, emr-6.0.0, emr-7.0.0

Kelas Instans	Tipe instans	Versi EMR Amazon yang didukung minimum (5.x, 6.x, 7.x)
	i3en.6xlarge	emr-5.25.0, emr-6.0.0, emr-7.0.0
	i3en.12xlarge	emr-5.25.0, emr-6.0.0, emr-7.0.0
	i3en.24xlarge	emr-5.25.0, emr-6.0.0, emr-7.0.0
	i4g.xlarge	emr-5.36.1, emr-6.10.0, emr-7.0.0
	i4g.2xlarge	emr-5.36.1, emr-6.10.0, emr-7.0.0
	i4g.4xlarge	emr-5.36.1, emr-6.10.0, emr-7.0.0
	i4g.8xlarge	emr-5.36.1, emr-6.10.0, emr-7.0.0
	i4g.16xlarge	emr-5.36.1, emr-6.10.0, emr-7.0.0
	i4i.xlarge	emr-5.36.1, emr-6.8.0
	i4i.2xlarge	emr-5.36.1, emr-6.8.0
	i4i.4xlarge	emr-5.36.1, emr-6.8.0
	i4i.8xlarge	emr-5.36.1, emr-6.8.0
	i4i.12xlarge	emr-5.36.1, emr-6.10.0
	i4i.16xlarge	emr-5.36.1, emr-6.8.0

Kelas Instans	Tipe instans	Versi EMR Amazon yang didukung minimum (5.x, 6.x, 7.x)
	i4i.24xlarge	emr-5.36.1, emr-6.10.0
	i4i.32xlarge	emr-5.36.1, emr-6.8.0
	im4gn.xlarge	emr-5.35.0, emr-6.6.0, emr-7.0.0
	im4gn.2xlarge	emr-5.35.0, emr-6.6.0, emr-7.0.0
	im4gn.4xlarge	emr-5.35.0, emr-6.6.0, emr-7.0.0
	im4gn.8xlarge	emr-5.35.0, emr-6.6.0, emr-7.0.0
	im4gn.16xlarge	emr-5.35.0, emr-6.6.0, emr-7.0.0
	is4gen.xlarge	emr-5.35.0, emr-6.6.0, emr-7.0.0
	is4gen.2xlarge	emr-5.35.0, emr-6.6.0, emr-7.0.0
	is4gen.4xlarge	emr-5.35.0, emr-6.6.0, emr-7.0.0
	is4gen.8xlarge	emr-5.35.0, emr-6.6.0, emr-7.0.0

AWS GovCloud (AS-Barat) - -1 us-gov-west

Kelas Instans	Tipe instans	Versi EMR Amazon yang didukung minimum (5.x, 6.x, 7.x)
Tujuan Umum	m5.xlarge	emr-5.13.0, emr-6.0.0, emr-7.0.0
	m5.2xlarge	emr-5.13.0, emr-6.0.0, emr-7.0.0
	m5.4xlarge	emr-5.13.0, emr-6.0.0, emr-7.0.0
	m5.8xlarge	emr-5.13.0, emr-6.0.0, emr-7.0.0
	m5.12xlarge	emr-5.13.0, emr-6.0.0, emr-7.0.0
	m5.16xlarge	emr-5.13.0, emr-6.0.0, emr-7.0.0
	m5.24xlarge	emr-5.13.0, emr-6.0.0, emr-7.0.0
	m5a.xlarge	emr-5.20.0, emr-6.0.0, emr-7.0.0
	m5a.2xlarge	emr-5.20.0, emr-6.0.0, emr-7.0.0
	m5a.4xlarge	emr-5.20.0, emr-6.0.0, emr-7.0.0
	m5a.8xlarge	emr-5.20.0, emr-6.0.0, emr-7.0.0

Kelas Instans	Tipe instans	Versi EMR Amazon yang didukung minimum (5.x, 6.x, 7.x)
	m5a.12xlarge	emr-5.20.0, emr-6.0.0, emr-7.0.0
	m5a.16xlarge	emr-5.20.0, emr-6.0.0, emr-7.0.0
	m5a.24xlarge	emr-5.20.0, emr-6.0.0, emr-7.0.0
	m5ad.xlarge	emr-5.20.0, emr-6.0.0, emr-7.0.0
	m5ad.2xlarge	emr-5.20.0, emr-6.0.0, emr-7.0.0
	m5ad.4xlarge	emr-5.20.0, emr-6.0.0, emr-7.0.0
	m5ad.8xlarge	emr-5.33.0, emr-6.3.0, emr-7.0.0
	m5ad.12xlarge	emr-5.20.0, emr-6.0.0, emr-7.0.0
	m5ad.16xlarge	emr-5.33.0, emr-6.3.0, emr-7.0.0
	m5ad.24xlarge	emr-5.20.0, emr-6.0.0, emr-7.0.0
	m5d.xlarge	emr-5.13.0, emr-6.0.0, emr-7.0.0
	m5d.2xlarge	emr-5.13.0, emr-6.0.0, emr-7.0.0

Kelas Instans	Tipe instans	Versi EMR Amazon yang didukung minimum (5.x, 6.x, 7.x)
	m5d.4xlarge	emr-5.13.0, emr-6.0.0, emr-7.0.0
	m5d.8xlarge	emr-5.13.0, emr-6.0.0, emr-7.0.0
	m5d.12xlarge	emr-5.13.0, emr-6.0.0, emr-7.0.0
	m5d.16xlarge	emr-5.13.0, emr-6.0.0, emr-7.0.0
	m5d.24xlarge	emr-5.13.0, emr-6.0.0, emr-7.0.0
	m5dn.xlarge	emr-5.31.0, emr-6.1.0, emr-7.0.0
	m5dn.2xlarge	emr-5.31.0, emr-6.1.0, emr-7.0.0
	m5dn.4xlarge	emr-5.31.0, emr-6.1.0, emr-7.0.0
	m5dn.8xlarge	emr-5.31.0, emr-6.1.0, emr-7.0.0
	m5dn.12xlarge	emr-5.31.0, emr-6.1.0, emr-7.0.0
	m5dn.16xlarge	emr-5.31.0, emr-6.1.0, emr-7.0.0
	m5dn.24xlarge	emr-5.31.0, emr-6.1.0, emr-7.0.0

Kelas Instans	Tipe instans	Versi EMR Amazon yang didukung minimum (5.x, 6.x, 7.x)
	m5n.xlarge	emr-5.31.0, emr-6.1.0, emr-7.0.0
	m5n.2xlarge	emr-5.31.0, emr-6.1.0, emr-7.0.0
	m5n.4xlarge	emr-5.31.0, emr-6.1.0, emr-7.0.0
	m5n.8xlarge	emr-5.31.0, emr-6.1.0, emr-7.0.0
	m5n.12xlarge	emr-5.31.0, emr-6.1.0, emr-7.0.0
	m5n.16xlarge	emr-5.31.0, emr-6.1.0, emr-7.0.0
	m5n.24xlarge	emr-5.31.0, emr-6.1.0, emr-7.0.0
	m6g.xlarge	emr-5.30.0, emr-6.1.0, emr-7.0.0
	m6g.2xlarge	emr-5.30.0, emr-6.1.0, emr-7.0.0
	m6g.4xlarge	emr-5.30.0, emr-6.1.0, emr-7.0.0
	m6g.8xlarge	emr-5.30.0, emr-6.1.0, emr-7.0.0
	m6g.12xlarge	emr-5.30.0, emr-6.1.0, emr-7.0.0

Kelas Instans	Tipe instans	Versi EMR Amazon yang didukung minimum (5.x, 6.x, 7.x)
	m6g.16xlarge	emr-5.30.0, emr-6.1.0, emr-7.0.0
	m6i.xlarge	emr-5.35.0, emr-6.6.0
	m6i.2xlarge	emr-5.35.0, emr-6.6.0
	m6i.4xlarge	emr-5.35.0, emr-6.6.0
	m6i.8xlarge	emr-5.35.0, emr-6.6.0
	m6i.12xlarge	emr-5.35.0, emr-6.6.0
	m6i.16xlarge	emr-5.35.0, emr-6.6.0
	m6i.24xlarge	emr-5.35.0, emr-6.6.0
	m6i.32xlarge	emr-5.35.0, emr-6.6.0
	m6id.xlarge	emr-5.36.1, emr-6.8.0
	m6id.2xlarge	emr-5.36.1, emr-6.8.0
	m6id.4xlarge	emr-5.36.1, emr-6.8.0
	m6id.8xlarge	emr-5.36.1, emr-6.8.0
	m6id.12xlarge	emr-5.36.1, emr-6.8.0
	m6id.16xlarge	emr-5.36.1, emr-6.8.0
	m6id.24xlarge	emr-5.36.1, emr-6.8.0
	m6id.32xlarge	emr-5.36.1, emr-6.8.0
	m6idn.xlarge	emr-5.36.1, emr-6.10.0
	m6idn.2xlarge	emr-5.36.1, emr-6.10.0

Kelas Instans	Tipe instans	Versi EMR Amazon yang didukung minimum (5.x, 6.x, 7.x)
	m6idn.4xlarge	emr-5.36.1, emr-6.10.0
	m6idn.8xlarge	emr-5.36.1, emr-6.10.0
	m6idn.12xlarge	emr-5.36.1, emr-6.10.0
	m6idn.16xlarge	emr-5.36.1, emr-6.10.0
	m6idn.24xlarge	emr-5.36.1, emr-6.10.0
	m6idn.32xlarge	emr-5.36.1, emr-6.10.0
	m6in.xlarge	emr-5.36.1, emr-6.10.0
	m6in.2xlarge	emr-5.36.1, emr-6.10.0
	m6in.4xlarge	emr-5.36.1, emr-6.10.0
	m6in.8xlarge	emr-5.36.1, emr-6.10.0
	m6in.12xlarge	emr-5.36.1, emr-6.10.0
	m6in.16xlarge	emr-5.36.1, emr-6.10.0
	m6in.24xlarge	emr-5.36.1, emr-6.10.0
	m6in.32xlarge	emr-5.36.1, emr-6.10.0
Komputasi Dioptimalkan	c5.xlarge	emr-5.13.0, emr-6.0.0, emr-7.0.0
	c5.2xlarge	emr-5.13.0, emr-6.0.0, emr-7.0.0
	c5.4xlarge	emr-5.13.0, emr-6.0.0, emr-7.0.0

Kelas Instans	Tipe instans	Versi EMR Amazon yang didukung minimum (5.x, 6.x, 7.x)
	c5.9xlarge	emr-5.13.0, emr-6.0.0, emr-7.0.0
	c5.12xlarge	emr-5.13.0, emr-6.0.0, emr-7.0.0
	c5.18xlarge	emr-5.13.0, emr-6.0.0, emr-7.0.0
	c5.24xlarge	emr-5.13.0, emr-6.0.0, emr-7.0.0
	c5a.xlarge	emr-5.31.0, emr-6.1.0, emr-7.0.0
	c5a.2xlarge	emr-5.31.0, emr-6.1.0, emr-7.0.0
	c5a.4xlarge	emr-5.31.0, emr-6.1.0, emr-7.0.0
	c5a.8xlarge	emr-5.31.0, emr-6.1.0, emr-7.0.0
	c5a.12xlarge	emr-5.31.0, emr-6.1.0, emr-7.0.0
	c5a.16xlarge	emr-5.31.0, emr-6.1.0, emr-7.0.0
	c5a.24xlarge	emr-5.31.0, emr-6.1.0, emr-7.0.0
	c5d.xlarge	emr-5.13.0, emr-6.0.0, emr-7.0.0

Kelas Instans	Tipe instans	Versi EMR Amazon yang didukung minimum (5.x, 6.x, 7.x)
	c5d.2xlarge	emr-5.13.0, emr-6.0.0, emr-7.0.0
	c5d.4xlarge	emr-5.13.0, emr-6.0.0, emr-7.0.0
	c5d.9xlarge	emr-5.13.0, emr-6.0.0, emr-7.0.0
	c5d.12xlarge	emr-5.13.0, emr-6.0.0, emr-7.0.0
	c5d.18xlarge	emr-5.13.0, emr-6.0.0, emr-7.0.0
	c5d.24xlarge	emr-5.13.0, emr-6.0.0, emr-7.0.0
	c5n.xlarge	emr-5.20.0, emr-6.0.0, emr-7.0.0
	c5n.2xlarge	emr-5.20.0, emr-6.0.0, emr-7.0.0
	c5n.4xlarge	emr-5.20.0, emr-6.0.0, emr-7.0.0
	c5n.9xlarge	emr-5.20.0, emr-6.0.0, emr-7.0.0
	c5n.18xlarge	emr-5.20.0, emr-6.0.0, emr-7.0.0
	c6g.xlarge	emr-5.31.0, emr-6.1.0, emr-7.0.0

Kelas Instans	Tipe instans	Versi EMR Amazon yang didukung minimum (5.x, 6.x, 7.x)
	c6g.2xlarge	emr-5.31.0, emr-6.1.0, emr-7.0.0
	c6g.4xlarge	emr-5.31.0, emr-6.1.0, emr-7.0.0
	c6g.8xlarge	emr-5.31.0, emr-6.1.0, emr-7.0.0
	c6g.12xlarge	emr-5.31.0, emr-6.1.0, emr-7.0.0
	c6g.16xlarge	emr-5.31.0, emr-6.1.0, emr-7.0.0
	c6i.xlarge	emr-5.35.0, emr-6.6.0
	c6i.2xlarge	emr-5.35.0, emr-6.6.0
	c6i.4xlarge	emr-5.35.0, emr-6.6.0
	c6i.8xlarge	emr-5.35.0, emr-6.6.0
	c6i.12xlarge	emr-5.35.0, emr-6.6.0
	c6i.16xlarge	emr-5.35.0, emr-6.6.0
	c6i.24xlarge	emr-5.35.0, emr-6.6.0
	c6i.32xlarge	emr-5.35.0, emr-6.6.0
	c6id.xlarge	emr-5.36.1, emr-6.8.0
	c6id.2xlarge	emr-5.36.1, emr-6.8.0
	c6id.4xlarge	emr-5.36.1, emr-6.8.0

Kelas Instans	Tipe instans	Versi EMR Amazon yang didukung minimum (5.x, 6.x, 7.x)
	c6id.8xlarge	emr-5.36.1, emr-6.8.0
	c6id.12xlarge	emr-5.36.1, emr-6.8.0
	c6id.16xlarge	emr-5.36.1, emr-6.8.0
	c6id.24xlarge	emr-5.36.1, emr-6.8.0
	c6id.32xlarge	emr-5.36.1, emr-6.8.0
	c6in.xlarge	emr-5.36.1, emr-6.10.0
	c6in.2xlarge	emr-5.36.1, emr-6.10.0
	c6in.4xlarge	emr-5.36.1, emr-6.10.0
	c6in.8xlarge	emr-5.36.1, emr-6.10.0
	c6in.12xlarge	emr-5.36.1, emr-6.10.0
	c6in.16xlarge	emr-5.36.1, emr-6.10.0
	c6in.24xlarge	emr-5.36.1, emr-6.10.0
	c6in.32xlarge	emr-5.36.1, emr-6.10.0
Komputasi yang Dipercepat	g3.4xlarge	emr-5.18.0, emr-6.0.0, emr-7.0.0
	g3.8xlarge	emr-5.18.0, emr-6.0.0, emr-7.0.0
	g3.16xlarge	emr-5.18.0, emr-6.0.0, emr-7.0.0
	g4dn.xlarge	emr-5.30.0, emr-6.0.0, emr-7.0.0

Kelas Instans	Tipe instans	Versi EMR Amazon yang didukung minimum (5.x, 6.x, 7.x)
	g4dn.2xlarge	emr-5.30.0, emr-6.0.0, emr-7.0.0
	g4dn.4xlarge	emr-5.30.0, emr-6.0.0, emr-7.0.0
	g4dn.8xlarge	emr-5.30.0, emr-6.0.0, emr-7.0.0
	g4dn.12xlarge	emr-5.30.0, emr-6.0.0, emr-7.0.0
	g4dn.16xlarge	emr-5.30.0, emr-6.0.0, emr-7.0.0
	p2.xlarge	emr-5.10.0, emr-6.0.0, emr-7.0.0
	p2.8xlarge	emr-5.10.0, emr-6.0.0, emr-7.0.0
	p2.16xlarge	emr-5.10.0, emr-6.0.0, emr-7.0.0
	p3.2xlarge	emr-5.10.0, emr-6.0.0, emr-7.0.0
	p3.8xlarge	emr-5.10.0, emr-6.0.0, emr-7.0.0
	p3.16xlarge	emr-5.10.0, emr-6.0.0, emr-7.0.0
Memori Dioptimalkan	r5.xlarge	emr-5.13.0, emr-6.0.0, emr-7.0.0

Kelas Instans	Tipe instans	Versi EMR Amazon yang didukung minimum (5.x, 6.x, 7.x)
	r5.2xlarge	emr-5.13.0, emr-6.0.0, emr-7.0.0
	r5.4xlarge	emr-5.13.0, emr-6.0.0, emr-7.0.0
	r5.8xlarge	emr-5.13.0, emr-6.0.0, emr-7.0.0
	r5.12xlarge	emr-5.13.0, emr-6.0.0, emr-7.0.0
	r5.16xlarge	emr-5.13.0, emr-6.0.0, emr-7.0.0
	r5.24xlarge	emr-5.13.0, emr-6.0.0, emr-7.0.0
	r5a.xlarge	emr-5.20.0, emr-6.0.0, emr-7.0.0
	r5a.2xlarge	emr-5.20.0, emr-6.0.0, emr-7.0.0
	r5a.4xlarge	emr-5.20.0, emr-6.0.0, emr-7.0.0
	r5a.8xlarge	emr-5.20.0, emr-6.0.0, emr-7.0.0
	r5a.12xlarge	emr-5.20.0, emr-6.0.0, emr-7.0.0
	r5a.16xlarge	emr-5.20.0, emr-6.0.0, emr-7.0.0

Kelas Instans	Tipe instans	Versi EMR Amazon yang didukung minimum (5.x, 6.x, 7.x)
	r5a.24xlarge	emr-5.20.0, emr-6.0.0, emr-7.0.0
	r5ad.xlarge	emr-5.20.0, emr-6.0.0, emr-7.0.0
	r5ad.2xlarge	emr-5.20.0, emr-6.0.0, emr-7.0.0
	r5ad.4xlarge	emr-5.20.0, emr-6.0.0, emr-7.0.0
	r5ad.8xlarge	emr-5.33.0, emr-6.3.0, emr-7.0.0
	r5ad.12xlarge	emr-5.20.0, emr-6.0.0, emr-7.0.0
	r5ad.16xlarge	emr-5.33.0, emr-6.3.0, emr-7.0.0
	r5ad.24xlarge	emr-5.20.0, emr-6.0.0, emr-7.0.0
	r5d.xlarge	emr-5.13.0, emr-6.0.0, emr-7.0.0
	r5d.2xlarge	emr-5.13.0, emr-6.0.0, emr-7.0.0
	r5d.4xlarge	emr-5.13.0, emr-6.0.0, emr-7.0.0
	r5d.8xlarge	emr-5.13.0, emr-6.0.0, emr-7.0.0

Kelas Instans	Tipe instans	Versi EMR Amazon yang didukung minimum (5.x, 6.x, 7.x)
	r5d.12xlarge	emr-5.13.0, emr-6.0.0, emr-7.0.0
	r5d.16xlarge	emr-5.13.0, emr-6.0.0, emr-7.0.0
	r5d.24xlarge	emr-5.13.0, emr-6.0.0, emr-7.0.0
	r5dn.xlarge	emr-5.31.0, emr-6.1.0, emr-7.0.0
	r5dn.2xlarge	emr-5.31.0, emr-6.1.0, emr-7.0.0
	r5dn.4xlarge	emr-5.31.0, emr-6.1.0, emr-7.0.0
	r5dn.8xlarge	emr-5.31.0, emr-6.1.0, emr-7.0.0
	r5dn.12xlarge	emr-5.31.0, emr-6.1.0, emr-7.0.0
	r5dn.16xlarge	emr-5.31.0, emr-6.1.0, emr-7.0.0
	r5dn.24xlarge	emr-5.31.0, emr-6.1.0, emr-7.0.0
	r5n.xlarge	emr-5.31.0, emr-6.1.0, emr-7.0.0
	r5n.2xlarge	emr-5.31.0, emr-6.1.0, emr-7.0.0

Kelas Instans	Tipe instans	Versi EMR Amazon yang didukung minimum (5.x, 6.x, 7.x)
	r5n.4xlarge	emr-5.31.0, emr-6.1.0, emr-7.0.0
	r5n.8xlarge	emr-5.31.0, emr-6.1.0, emr-7.0.0
	r5n.12xlarge	emr-5.31.0, emr-6.1.0, emr-7.0.0
	r5n.16xlarge	emr-5.31.0, emr-6.1.0, emr-7.0.0
	r5n.24xlarge	emr-5.31.0, emr-6.1.0, emr-7.0.0
	r6g.xlarge	emr-5.31.0, emr-6.1.0, emr-7.0.0
	r6g.2xlarge	emr-5.31.0, emr-6.1.0, emr-7.0.0
	r6g.4xlarge	emr-5.31.0, emr-6.1.0, emr-7.0.0
	r6g.8xlarge	emr-5.31.0, emr-6.1.0, emr-7.0.0
	r6g.12xlarge	emr-5.31.0, emr-6.1.0, emr-7.0.0
	r6g.16xlarge	emr-5.31.0, emr-6.1.0, emr-7.0.0
	r6i.xlarge	emr-5.35.0, emr-6.6.0
	r6i.2xlarge	emr-5.35.0, emr-6.6.0

Kelas Instans	Tipe instans	Versi EMR Amazon yang didukung minimum (5.x, 6.x, 7.x)
	r6i.4xlarge	emr-5.35.0, emr-6.6.0
	r6i.8xlarge	emr-5.35.0, emr-6.6.0
	r6i.12xlarge	emr-5.35.0, emr-6.6.0
	r6i.16xlarge	emr-5.35.0, emr-6.6.0
	r6i.24xlarge	emr-5.35.0, emr-6.6.0
	r6i.32xlarge	emr-5.35.0, emr-6.6.0
	r6id.xlarge	emr-5.36.1, emr-6.8.0
	r6id.2xlarge	emr-5.36.1, emr-6.8.0
	r6id.4xlarge	emr-5.36.1, emr-6.8.0
	r6id.8xlarge	emr-5.36.1, emr-6.8.0
	r6id.12xlarge	emr-5.36.1, emr-6.8.0
	r6id.16xlarge	emr-5.36.1, emr-6.8.0
	r6id.24xlarge	emr-5.36.1, emr-6.8.0
	r6id.32xlarge	emr-5.36.1, emr-6.8.0
	r6idn.xlarge	emr-5.36.1, emr-6.10.0
	r6idn.2xlarge	emr-5.36.1, emr-6.10.0
	r6idn.4xlarge	emr-5.36.1, emr-6.10.0
	r6idn.8xlarge	emr-5.36.1, emr-6.10.0
	r6idn.12xlarge	emr-5.36.1, emr-6.10.0

Kelas Instans	Tipe instans	Versi EMR Amazon yang didukung minimum (5.x, 6.x, 7.x)
	r6idn.16xlarge	emr-5.36.1, emr-6.10.0
	r6idn.24xlarge	emr-5.36.1, emr-6.10.0
	r6idn.32xlarge	emr-5.36.1, emr-6.10.0
	r6in.xlarge	emr-5.36.1, emr-6.10.0
	r6in.2xlarge	emr-5.36.1, emr-6.10.0
	r6in.4xlarge	emr-5.36.1, emr-6.10.0
	r6in.8xlarge	emr-5.36.1, emr-6.10.0
	r6in.12xlarge	emr-5.36.1, emr-6.10.0
	r6in.16xlarge	emr-5.36.1, emr-6.10.0
	r6in.24xlarge	emr-5.36.1, emr-6.10.0
	r6in.32xlarge	emr-5.36.1, emr-6.10.0
	x1.16xlarge	emr-5.36.1, emr-6.9.0, emr-7.0.0
	x1.32xlarge	emr-5.36.1, emr-6.9.0, emr-7.0.0
	x1e.xlarge	emr-5.36.1, emr-6.10.0, emr-7.0.0
	x1e.2xlarge	emr-5.36.1, emr-6.10.0, emr-7.0.0
	x1e.4xlarge	emr-5.36.1, emr-6.10.0, emr-7.0.0

Kelas Instans	Tipe instans	Versi EMR Amazon yang didukung minimum (5.x, 6.x, 7.x)
	x1e.8xlarge	emr-5.36.1, emr-6.10.0, emr-7.0.0
	x1e.16xlarge	emr-5.36.1, emr-6.10.0, emr-7.0.0
	x1e.32xlarge	emr-5.36.1, emr-6.10.0, emr-7.0.0
	x2idn.16xlarge	emr-5.36.1, emr-6.9.0
	x2idn.24xlarge	emr-5.36.1, emr-6.9.0
	x2idn.32xlarge	emr-5.36.1, emr-6.9.0
	x2iedn.xlarge	emr-5.36.1, emr-6.9.0
	x2iedn.2xlarge	emr-5.36.1, emr-6.9.0
	x2iedn.4xlarge	emr-5.36.1, emr-6.9.0
	x2iedn.8xlarge	emr-5.36.1, emr-6.9.0
	x2iedn.16xlarge	emr-5.36.1, emr-6.9.0
	x2iedn.24xlarge	emr-5.36.1, emr-6.9.0
	x2iedn.32xlarge	emr-5.36.1, emr-6.9.0
Penyimpanan Dioptimalkan	d3.xlarge	emr-5.33.0, emr-6.3.0, emr-7.0.0
	d3.2xlarge	emr-5.33.0, emr-6.3.0, emr-7.0.0
	d3.4xlarge	emr-5.33.0, emr-6.3.0, emr-7.0.0

Kelas Instans	Tipe instans	Versi EMR Amazon yang didukung minimum (5.x, 6.x, 7.x)
	d3.8xlarge	emr-5.33.0, emr-6.3.0, emr-7.0.0
	i3.xlarge	emr-5.9.0, emr-6.0.0, emr-7.0.0
	i3.2xlarge	emr-5.9.0, emr-6.0.0, emr-7.0.0
	i3.4xlarge	emr-5.9.0, emr-6.0.0, emr-7.0.0
	i3.8xlarge	emr-5.9.0, emr-6.0.0, emr-7.0.0
	i3.16xlarge	emr-5.9.0, emr-6.0.0, emr-7.0.0
	i3en.xlarge	emr-5.25.0, emr-6.0.0, emr-7.0.0
	i3en.2xlarge	emr-5.25.0, emr-6.0.0, emr-7.0.0
	i3en.3xlarge	emr-5.25.0, emr-6.0.0, emr-7.0.0
	i3en.6xlarge	emr-5.25.0, emr-6.0.0, emr-7.0.0
	i3en.12xlarge	emr-5.25.0, emr-6.0.0, emr-7.0.0
	i3en.24xlarge	emr-5.25.0, emr-6.0.0, emr-7.0.0

Kelas Instans	Tipe instans	Versi EMR Amazon yang didukung minimum (5.x, 6.x, 7.x)
	i4i.xlarge	emr-5.36.1, emr-6.8.0
	i4i.2xlarge	emr-5.36.1, emr-6.8.0
	i4i.4xlarge	emr-5.36.1, emr-6.8.0
	i4i.8xlarge	emr-5.36.1, emr-6.8.0
	i4i.16xlarge	emr-5.36.1, emr-6.8.0
	i4i.32xlarge	emr-5.36.1, emr-6.8.0

AWS GovCloud (AS-Timur) - -1 us-gov-east

Kelas Instans	Tipe instans	Versi EMR Amazon yang didukung minimum (5.x, 6.x, 7.x)
Tujuan Umum	m5.xlarge	emr-5.16.0, emr-6.0.0, emr-7.0.0
	m5.2xlarge	emr-5.16.0, emr-6.0.0, emr-7.0.0
	m5.4xlarge	emr-5.16.0, emr-6.0.0, emr-7.0.0
	m5.8xlarge	emr-5.16.0, emr-6.0.0, emr-7.0.0
	m5.12xlarge	emr-5.16.0, emr-6.0.0, emr-7.0.0
	m5.16xlarge	emr-5.16.0, emr-6.0.0, emr-7.0.0

Kelas Instans	Tipe instans	Versi EMR Amazon yang didukung minimum (5.x, 6.x, 7.x)
	m5.24xlarge	emr-5.16.0, emr-6.0.0, emr-7.0.0
	m5a.xlarge	emr-5.20.0, emr-6.0.0, emr-7.0.0
	m5a.2xlarge	emr-5.20.0, emr-6.0.0, emr-7.0.0
	m5a.4xlarge	emr-5.20.0, emr-6.0.0, emr-7.0.0
	m5a.8xlarge	emr-5.20.0, emr-6.0.0, emr-7.0.0
	m5a.12xlarge	emr-5.20.0, emr-6.0.0, emr-7.0.0
	m5a.16xlarge	emr-5.20.0, emr-6.0.0, emr-7.0.0
	m5a.24xlarge	emr-5.20.0, emr-6.0.0, emr-7.0.0
	m5d.xlarge	emr-5.16.0, emr-6.0.0, emr-7.0.0
	m5d.2xlarge	emr-5.16.0, emr-6.0.0, emr-7.0.0
	m5d.4xlarge	emr-5.16.0, emr-6.0.0, emr-7.0.0
	m5d.8xlarge	emr-5.16.0, emr-6.0.0, emr-7.0.0

Kelas Instans	Tipe instans	Versi EMR Amazon yang didukung minimum (5.x, 6.x, 7.x)
	m5d.12xlarge	emr-5.16.0, emr-6.0.0, emr-7.0.0
	m5d.16xlarge	emr-5.16.0, emr-6.0.0, emr-7.0.0
	m5d.24xlarge	emr-5.16.0, emr-6.0.0, emr-7.0.0
	m5dn.xlarge	emr-5.31.0, emr-6.1.0, emr-7.0.0
	m5dn.2xlarge	emr-5.31.0, emr-6.1.0, emr-7.0.0
	m5dn.4xlarge	emr-5.31.0, emr-6.1.0, emr-7.0.0
	m5dn.8xlarge	emr-5.31.0, emr-6.1.0, emr-7.0.0
	m5dn.12xlarge	emr-5.31.0, emr-6.1.0, emr-7.0.0
	m5dn.16xlarge	emr-5.31.0, emr-6.1.0, emr-7.0.0
	m5dn.24xlarge	emr-5.31.0, emr-6.1.0, emr-7.0.0
	m5n.xlarge	emr-5.31.0, emr-6.1.0, emr-7.0.0
	m5n.2xlarge	emr-5.31.0, emr-6.1.0, emr-7.0.0

Kelas Instans	Tipe instans	Versi EMR Amazon yang didukung minimum (5.x, 6.x, 7.x)
	m5n.4xlarge	emr-5.31.0, emr-6.1.0, emr-7.0.0
	m5n.8xlarge	emr-5.31.0, emr-6.1.0, emr-7.0.0
	m5n.12xlarge	emr-5.31.0, emr-6.1.0, emr-7.0.0
	m5n.16xlarge	emr-5.31.0, emr-6.1.0, emr-7.0.0
	m5n.24xlarge	emr-5.31.0, emr-6.1.0, emr-7.0.0
	m6g.xlarge	emr-5.30.0, emr-6.1.0, emr-7.0.0
	m6g.2xlarge	emr-5.30.0, emr-6.1.0, emr-7.0.0
	m6g.4xlarge	emr-5.30.0, emr-6.1.0, emr-7.0.0
	m6g.8xlarge	emr-5.30.0, emr-6.1.0, emr-7.0.0
	m6g.12xlarge	emr-5.30.0, emr-6.1.0, emr-7.0.0
	m6g.16xlarge	emr-5.30.0, emr-6.1.0, emr-7.0.0
	m6i.xlarge	emr-5.35.0, emr-6.6.0
	m6i.2xlarge	emr-5.35.0, emr-6.6.0

Kelas Instans	Tipe instans	Versi EMR Amazon yang didukung minimum (5.x, 6.x, 7.x)
	m6i.4xlarge	emr-5.35.0, emr-6.6.0
	m6i.8xlarge	emr-5.35.0, emr-6.6.0
	m6i.12xlarge	emr-5.35.0, emr-6.6.0
	m6i.16xlarge	emr-5.35.0, emr-6.6.0
	m6i.24xlarge	emr-5.35.0, emr-6.6.0
	m6i.32xlarge	emr-5.35.0, emr-6.6.0
Komputasi Dioptimalkan	c5.xlarge	emr-5.16.0, emr-6.0.0, emr-7.0.0
	c5.2xlarge	emr-5.16.0, emr-6.0.0, emr-7.0.0
	c5.4xlarge	emr-5.16.0, emr-6.0.0, emr-7.0.0
	c5.9xlarge	emr-5.16.0, emr-6.0.0, emr-7.0.0
	c5.12xlarge	emr-5.16.0, emr-6.0.0, emr-7.0.0
	c5.18xlarge	emr-5.16.0, emr-6.0.0, emr-7.0.0
	c5.24xlarge	emr-5.16.0, emr-6.0.0, emr-7.0.0
	c5a.xlarge	emr-5.31.0, emr-6.1.0, emr-7.0.0

Kelas Instans	Tipe instans	Versi EMR Amazon yang didukung minimum (5.x, 6.x, 7.x)
	c5a.2xlarge	emr-5.31.0, emr-6.1.0, emr-7.0.0
	c5a.4xlarge	emr-5.31.0, emr-6.1.0, emr-7.0.0
	c5a.8xlarge	emr-5.31.0, emr-6.1.0, emr-7.0.0
	c5a.12xlarge	emr-5.31.0, emr-6.1.0, emr-7.0.0
	c5a.16xlarge	emr-5.31.0, emr-6.1.0, emr-7.0.0
	c5a.24xlarge	emr-5.31.0, emr-6.1.0, emr-7.0.0
	c5d.xlarge	emr-5.16.0, emr-6.0.0, emr-7.0.0
	c5d.2xlarge	emr-5.16.0, emr-6.0.0, emr-7.0.0
	c5d.4xlarge	emr-5.16.0, emr-6.0.0, emr-7.0.0
	c5d.9xlarge	emr-5.16.0, emr-6.0.0, emr-7.0.0
	c5d.18xlarge	emr-5.16.0, emr-6.0.0, emr-7.0.0
	c5n.xlarge	emr-5.20.0, emr-6.0.0, emr-7.0.0

Kelas Instans	Tipe instans	Versi EMR Amazon yang didukung minimum (5.x, 6.x, 7.x)
	c5n.2xlarge	emr-5.20.0, emr-6.0.0, emr-7.0.0
	c5n.4xlarge	emr-5.20.0, emr-6.0.0, emr-7.0.0
	c5n.9xlarge	emr-5.20.0, emr-6.0.0, emr-7.0.0
	c5n.18xlarge	emr-5.20.0, emr-6.0.0, emr-7.0.0
	c6g.xlarge	emr-5.31.0, emr-6.1.0, emr-7.0.0
	c6g.2xlarge	emr-5.31.0, emr-6.1.0, emr-7.0.0
	c6g.4xlarge	emr-5.31.0, emr-6.1.0, emr-7.0.0
	c6g.8xlarge	emr-5.31.0, emr-6.1.0, emr-7.0.0
	c6g.12xlarge	emr-5.31.0, emr-6.1.0, emr-7.0.0
	c6g.16xlarge	emr-5.31.0, emr-6.1.0, emr-7.0.0
	c6i.xlarge	emr-5.35.0, emr-6.6.0
	c6i.2xlarge	emr-5.35.0, emr-6.6.0
	c6i.4xlarge	emr-5.35.0, emr-6.6.0

Kelas Instans	Tipe instans	Versi EMR Amazon yang didukung minimum (5.x, 6.x, 7.x)
	c6i.8xlarge	emr-5.35.0, emr-6.6.0
	c6i.12xlarge	emr-5.35.0, emr-6.6.0
	c6i.16xlarge	emr-5.35.0, emr-6.6.0
	c6i.24xlarge	emr-5.35.0, emr-6.6.0
	c6i.32xlarge	emr-5.35.0, emr-6.6.0
	c6in.xlarge	emr-5.36.1, emr-6.10.0
	c6in.2xlarge	emr-5.36.1, emr-6.10.0
	c6in.4xlarge	emr-5.36.1, emr-6.10.0
	c6in.8xlarge	emr-5.36.1, emr-6.10.0
	c6in.12xlarge	emr-5.36.1, emr-6.10.0
	c6in.16xlarge	emr-5.36.1, emr-6.10.0
	c6in.24xlarge	emr-5.36.1, emr-6.10.0
	c6in.32xlarge	emr-5.36.1, emr-6.10.0
Komputasi yang Dipercepat	g4dn.xlarge	emr-5.30.0, emr-6.0.0, emr-7.0.0
	g4dn.2xlarge	emr-5.30.0, emr-6.0.0, emr-7.0.0
	g4dn.4xlarge	emr-5.30.0, emr-6.0.0, emr-7.0.0
	g4dn.8xlarge	emr-5.30.0, emr-6.0.0, emr-7.0.0

Kelas Instans	Tipe instans	Versi EMR Amazon yang didukung minimum (5.x, 6.x, 7.x)
	g4dn.12xlarge	emr-5.30.0, emr-6.0.0, emr-7.0.0
	g4dn.16xlarge	emr-5.30.0, emr-6.0.0, emr-7.0.0
Memori Dioptimalkan	r5.xlarge	emr-5.16.0, emr-6.0.0, emr-7.0.0
	r5.2xlarge	emr-5.16.0, emr-6.0.0, emr-7.0.0
	r5.4xlarge	emr-5.16.0, emr-6.0.0, emr-7.0.0
	r5.8xlarge	emr-5.16.0, emr-6.0.0, emr-7.0.0
	r5.12xlarge	emr-5.16.0, emr-6.0.0, emr-7.0.0
	r5.16xlarge	emr-5.16.0, emr-6.0.0, emr-7.0.0
	r5.24xlarge	emr-5.16.0, emr-6.0.0, emr-7.0.0
	r5a.xlarge	emr-5.20.0, emr-6.0.0, emr-7.0.0
	r5a.2xlarge	emr-5.20.0, emr-6.0.0, emr-7.0.0
	r5a.4xlarge	emr-5.20.0, emr-6.0.0, emr-7.0.0

Kelas Instans	Tipe instans	Versi EMR Amazon yang didukung minimum (5.x, 6.x, 7.x)
	r5a.8xlarge	emr-5.20.0, emr-6.0.0, emr-7.0.0
	r5a.12xlarge	emr-5.20.0, emr-6.0.0, emr-7.0.0
	r5a.16xlarge	emr-5.20.0, emr-6.0.0, emr-7.0.0
	r5a.24xlarge	emr-5.20.0, emr-6.0.0, emr-7.0.0
	r5d.xlarge	emr-5.16.0, emr-6.0.0, emr-7.0.0
	r5d.2xlarge	emr-5.16.0, emr-6.0.0, emr-7.0.0
	r5d.4xlarge	emr-5.16.0, emr-6.0.0, emr-7.0.0
	r5d.8xlarge	emr-5.16.0, emr-6.0.0, emr-7.0.0
	r5d.12xlarge	emr-5.16.0, emr-6.0.0, emr-7.0.0
	r5d.16xlarge	emr-5.16.0, emr-6.0.0, emr-7.0.0
	r5d.24xlarge	emr-5.16.0, emr-6.0.0, emr-7.0.0
	r5dn.xlarge	emr-5.31.0, emr-6.1.0, emr-7.0.0

Kelas Instans	Tipe instans	Versi EMR Amazon yang didukung minimum (5.x, 6.x, 7.x)
	r5dn.2xlarge	emr-5.31.0, emr-6.1.0, emr-7.0.0
	r5dn.4xlarge	emr-5.31.0, emr-6.1.0, emr-7.0.0
	r5dn.8xlarge	emr-5.31.0, emr-6.1.0, emr-7.0.0
	r5dn.12xlarge	emr-5.31.0, emr-6.1.0, emr-7.0.0
	r5dn.16xlarge	emr-5.31.0, emr-6.1.0, emr-7.0.0
	r5dn.24xlarge	emr-5.31.0, emr-6.1.0, emr-7.0.0
	r5n.xlarge	emr-5.31.0, emr-6.1.0, emr-7.0.0
	r5n.2xlarge	emr-5.31.0, emr-6.1.0, emr-7.0.0
	r5n.4xlarge	emr-5.31.0, emr-6.1.0, emr-7.0.0
	r5n.8xlarge	emr-5.31.0, emr-6.1.0, emr-7.0.0
	r5n.12xlarge	emr-5.31.0, emr-6.1.0, emr-7.0.0
	r5n.16xlarge	emr-5.31.0, emr-6.1.0, emr-7.0.0

Kelas Instans	Tipe instans	Versi EMR Amazon yang didukung minimum (5.x, 6.x, 7.x)
	r5n.24xlarge	emr-5.31.0, emr-6.1.0, emr-7.0.0
	r6g.xlarge	emr-5.31.0, emr-6.1.0, emr-7.0.0
	r6g.2xlarge	emr-5.31.0, emr-6.1.0, emr-7.0.0
	r6g.4xlarge	emr-5.31.0, emr-6.1.0, emr-7.0.0
	r6g.8xlarge	emr-5.31.0, emr-6.1.0, emr-7.0.0
	r6g.12xlarge	emr-5.31.0, emr-6.1.0, emr-7.0.0
	r6g.16xlarge	emr-5.31.0, emr-6.1.0, emr-7.0.0
	r6i.xlarge	emr-5.35.0, emr-6.6.0
	r6i.2xlarge	emr-5.35.0, emr-6.6.0
	r6i.4xlarge	emr-5.35.0, emr-6.6.0
	r6i.8xlarge	emr-5.35.0, emr-6.6.0
	r6i.12xlarge	emr-5.35.0, emr-6.6.0
	r6i.16xlarge	emr-5.35.0, emr-6.6.0
	r6i.24xlarge	emr-5.35.0, emr-6.6.0
	r6i.32xlarge	emr-5.35.0, emr-6.6.0

Kelas Instans	Tipe instans	Versi EMR Amazon yang didukung minimum (5.x, 6.x, 7.x)
	x1.16xlarge	emr-5.36.1, emr-6.9.0, emr-7.0.0
	x1.32xlarge	emr-5.36.1, emr-6.9.0, emr-7.0.0
	x1e.xlarge	emr-5.36.1, emr-6.10.0, emr-7.0.0
	x1e.2xlarge	emr-5.36.1, emr-6.10.0, emr-7.0.0
	x1e.4xlarge	emr-5.36.1, emr-6.10.0, emr-7.0.0
	x1e.8xlarge	emr-5.36.1, emr-6.10.0, emr-7.0.0
	x1e.16xlarge	emr-5.36.1, emr-6.10.0, emr-7.0.0
	x1e.32xlarge	emr-5.36.1, emr-6.10.0, emr-7.0.0
	x2idn.16xlarge	emr-5.36.1, emr-6.9.0
	x2idn.24xlarge	emr-5.36.1, emr-6.9.0
	x2idn.32xlarge	emr-5.36.1, emr-6.9.0
	x2iedn.xlarge	emr-5.36.1, emr-6.9.0
	x2iedn.2xlarge	emr-5.36.1, emr-6.9.0
	x2iedn.4xlarge	emr-5.36.1, emr-6.9.0

Kelas Instans	Tipe instans	Versi EMR Amazon yang didukung minimum (5.x, 6.x, 7.x)
	x2iedn.8xlarge	emr-5.36.1, emr-6.9.0
	x2iedn.16xlarge	emr-5.36.1, emr-6.9.0
	x2iedn.24xlarge	emr-5.36.1, emr-6.9.0
	x2iedn.32xlarge	emr-5.36.1, emr-6.9.0
Penyimpanan Dioptimalkan	i3.xlarge	emr-5.16.0, emr-6.0.0, emr-7.0.0
	i3.2xlarge	emr-5.16.0, emr-6.0.0, emr-7.0.0
	i3.4xlarge	emr-5.16.0, emr-6.0.0, emr-7.0.0
	i3.8xlarge	emr-5.16.0, emr-6.0.0, emr-7.0.0
	i3.16xlarge	emr-5.16.0, emr-6.0.0, emr-7.0.0
	i3en.xlarge	emr-5.25.0, emr-6.0.0, emr-7.0.0
	i3en.2xlarge	emr-5.25.0, emr-6.0.0, emr-7.0.0
	i3en.3xlarge	emr-5.25.0, emr-6.0.0, emr-7.0.0
	i3en.6xlarge	emr-5.25.0, emr-6.0.0, emr-7.0.0

Kelas Instans	Tipe instans	Versi EMR Amazon yang didukung minimum (5.x, 6.x, 7.x)
	i3en.12xlarge	emr-5.25.0, emr-6.0.0, emr-7.0.0
	i3en.24xlarge	emr-5.25.0, emr-6.0.0, emr-7.0.0
	i4i.xlarge	emr-5.36.1, emr-6.8.0
	i4i.2xlarge	emr-5.36.1, emr-6.8.0
	i4i.4xlarge	emr-5.36.1, emr-6.8.0
	i4i.8xlarge	emr-5.36.1, emr-6.8.0
	i4i.16xlarge	emr-5.36.1, emr-6.8.0
	i4i.32xlarge	emr-5.36.1, emr-6.8.0

Afrika (Cape Town) - af-selatan-1

Kelas Instans	Tipe instans	Versi EMR Amazon yang didukung minimum (5.x, 6.x, 7.x)
Tujuan Umum	m5.xlarge	emr-5.29.0, emr-6.0.0, emr-7.0.0
	m5.2xlarge	emr-5.29.0, emr-6.0.0, emr-7.0.0
	m5.4xlarge	emr-5.29.0, emr-6.0.0, emr-7.0.0
	m5.8xlarge	emr-5.29.0, emr-6.0.0, emr-7.0.0

Kelas Instans	Tipe instans	Versi EMR Amazon yang didukung minimum (5.x, 6.x, 7.x)
	m5.12xlarge	emr-5.29.0, emr-6.0.0, emr-7.0.0
	m5.16xlarge	emr-5.29.0, emr-6.0.0, emr-7.0.0
	m5.24xlarge	emr-5.29.0, emr-6.0.0, emr-7.0.0
	m5d.xlarge	emr-5.29.0, emr-6.0.0, emr-7.0.0
	m5d.2xlarge	emr-5.29.0, emr-6.0.0, emr-7.0.0
	m5d.4xlarge	emr-5.29.0, emr-6.0.0, emr-7.0.0
	m5d.8xlarge	emr-5.29.0, emr-6.0.0, emr-7.0.0
	m5d.12xlarge	emr-5.29.0, emr-6.0.0, emr-7.0.0
	m5d.16xlarge	emr-5.29.0, emr-6.0.0, emr-7.0.0
	m5d.24xlarge	emr-5.29.0, emr-6.0.0, emr-7.0.0
	m6g.xlarge	emr-5.30.0, emr-6.1.0, emr-7.0.0
	m6g.2xlarge	emr-5.30.0, emr-6.1.0, emr-7.0.0

Kelas Instans	Tipe instans	Versi EMR Amazon yang didukung minimum (5.x, 6.x, 7.x)
	m6g.4xlarge	emr-5.30.0, emr-6.1.0, emr-7.0.0
	m6g.8xlarge	emr-5.30.0, emr-6.1.0, emr-7.0.0
	m6g.12xlarge	emr-5.30.0, emr-6.1.0, emr-7.0.0
	m6g.16xlarge	emr-5.30.0, emr-6.1.0, emr-7.0.0
	m6gd.xlarge	emr-5.33.0, emr-6.3.0, emr-7.0.0
	m6gd.2xlarge	emr-5.33.0, emr-6.3.0, emr-7.0.0
	m6gd.4xlarge	emr-5.33.0, emr-6.3.0, emr-7.0.0
	m6gd.8xlarge	emr-5.33.0, emr-6.3.0, emr-7.0.0
	m6gd.12xlarge	emr-5.33.0, emr-6.3.0, emr-7.0.0
	m6gd.16xlarge	emr-5.33.0, emr-6.3.0, emr-7.0.0
	m6i.xlarge	emr-5.35.0, emr-6.6.0
	m6i.2xlarge	emr-5.35.0, emr-6.6.0
	m6i.4xlarge	emr-5.35.0, emr-6.6.0

Kelas Instans	Tipe instans	Versi EMR Amazon yang didukung minimum (5.x, 6.x, 7.x)
	m6i.8xlarge	emr-5.35.0, emr-6.6.0
	m6i.12xlarge	emr-5.35.0, emr-6.6.0
	m6i.16xlarge	emr-5.35.0, emr-6.6.0
	m6i.24xlarge	emr-5.35.0, emr-6.6.0
	m6i.32xlarge	emr-5.35.0, emr-6.6.0
Komputasi Dioptimalkan	c5.xlarge	emr-5.29.0, emr-6.0.0, emr-7.0.0
	c5.2xlarge	emr-5.29.0, emr-6.0.0, emr-7.0.0
	c5.4xlarge	emr-5.29.0, emr-6.0.0, emr-7.0.0
	c5.9xlarge	emr-5.29.0, emr-6.0.0, emr-7.0.0
	c5.12xlarge	emr-5.29.0, emr-6.0.0, emr-7.0.0
	c5.18xlarge	emr-5.29.0, emr-6.0.0, emr-7.0.0
	c5.24xlarge	emr-5.29.0, emr-6.0.0, emr-7.0.0
	c5a.xlarge	emr-5.31.0, emr-6.1.0, emr-7.0.0
	c5a.2xlarge	emr-5.31.0, emr-6.1.0, emr-7.0.0

Kelas Instans	Tipe instans	Versi EMR Amazon yang didukung minimum (5.x, 6.x, 7.x)
	c5a.4xlarge	emr-5.31.0, emr-6.1.0, emr-7.0.0
	c5a.8xlarge	emr-5.31.0, emr-6.1.0, emr-7.0.0
	c5a.12xlarge	emr-5.31.0, emr-6.1.0, emr-7.0.0
	c5a.16xlarge	emr-5.31.0, emr-6.1.0, emr-7.0.0
	c5a.24xlarge	emr-5.31.0, emr-6.1.0, emr-7.0.0
	c5ad.xlarge	emr-5.33.0, emr-6.3.0, emr-7.0.0
	c5ad.2xlarge	emr-5.33.0, emr-6.3.0, emr-7.0.0
	c5ad.4xlarge	emr-5.33.0, emr-6.3.0, emr-7.0.0
	c5ad.8xlarge	emr-5.33.0, emr-6.3.0, emr-7.0.0
	c5ad.12xlarge	emr-5.33.0, emr-6.3.0, emr-7.0.0
	c5ad.16xlarge	emr-5.33.0, emr-6.3.0, emr-7.0.0
	c5ad.24xlarge	emr-5.33.0, emr-6.3.0, emr-7.0.0

Kelas Instans	Tipe instans	Versi EMR Amazon yang didukung minimum (5.x, 6.x, 7.x)
	c5d.xlarge	emr-5.29.0, emr-6.0.0, emr-7.0.0
	c5d.2xlarge	emr-5.29.0, emr-6.0.0, emr-7.0.0
	c5d.4xlarge	emr-5.29.0, emr-6.0.0, emr-7.0.0
	c5d.9xlarge	emr-5.29.0, emr-6.0.0, emr-7.0.0
	c5d.12xlarge	emr-5.29.0, emr-6.0.0, emr-7.0.0
	c5d.18xlarge	emr-5.29.0, emr-6.0.0, emr-7.0.0
	c5d.24xlarge	emr-5.29.0, emr-6.0.0, emr-7.0.0
	c5n.xlarge	emr-5.29.0, emr-6.0.0, emr-7.0.0
	c5n.2xlarge	emr-5.29.0, emr-6.0.0, emr-7.0.0
	c5n.4xlarge	emr-5.29.0, emr-6.0.0, emr-7.0.0
	c5n.9xlarge	emr-5.29.0, emr-6.0.0, emr-7.0.0
	c5n.18xlarge	emr-5.29.0, emr-6.0.0, emr-7.0.0

Kelas Instans	Tipe instans	Versi EMR Amazon yang didukung minimum (5.x, 6.x, 7.x)
	c6i.xlarge	emr-5.35.0, emr-6.6.0
	c6i.2xlarge	emr-5.35.0, emr-6.6.0
	c6i.4xlarge	emr-5.35.0, emr-6.6.0
	c6i.8xlarge	emr-5.35.0, emr-6.6.0
	c6i.12xlarge	emr-5.35.0, emr-6.6.0
	c6i.16xlarge	emr-5.35.0, emr-6.6.0
	c6i.24xlarge	emr-5.35.0, emr-6.6.0
	c6i.32xlarge	emr-5.35.0, emr-6.6.0
	c6in.xlarge	emr-5.36.1, emr-6.10.0
	c6in.2xlarge	emr-5.36.1, emr-6.10.0
	c6in.4xlarge	emr-5.36.1, emr-6.10.0
	c6in.8xlarge	emr-5.36.1, emr-6.10.0
	c6in.12xlarge	emr-5.36.1, emr-6.10.0
	c6in.16xlarge	emr-5.36.1, emr-6.10.0
	c6in.24xlarge	emr-5.36.1, emr-6.10.0
	c6in.32xlarge	emr-5.36.1, emr-6.10.0
Komputasi yang Dipercepat	g4dn.xlarge	emr-5.30.0, emr-6.0.0, emr-7.0.0
	g4dn.2xlarge	emr-5.30.0, emr-6.0.0, emr-7.0.0

Kelas Instans	Tipe instans	Versi EMR Amazon yang didukung minimum (5.x, 6.x, 7.x)
	g4dn.4xlarge	emr-5.30.0, emr-6.0.0, emr-7.0.0
	g4dn.8xlarge	emr-5.30.0, emr-6.0.0, emr-7.0.0
	g4dn.12xlarge	emr-5.30.0, emr-6.0.0, emr-7.0.0
	g4dn.16xlarge	emr-5.30.0, emr-6.0.0, emr-7.0.0
Memori Dioptimalkan	r5.xlarge	emr-5.29.0, emr-6.0.0, emr-7.0.0
	r5.2xlarge	emr-5.29.0, emr-6.0.0, emr-7.0.0
	r5.4xlarge	emr-5.29.0, emr-6.0.0, emr-7.0.0
	r5.8xlarge	emr-5.29.0, emr-6.0.0, emr-7.0.0
	r5.12xlarge	emr-5.29.0, emr-6.0.0, emr-7.0.0
	r5.16xlarge	emr-5.29.0, emr-6.0.0, emr-7.0.0
	r5.24xlarge	emr-5.29.0, emr-6.0.0, emr-7.0.0
	r5d.xlarge	emr-5.29.0, emr-6.0.0, emr-7.0.0

Kelas Instans	Tipe instans	Versi EMR Amazon yang didukung minimum (5.x, 6.x, 7.x)
	r5d.2xlarge	emr-5.29.0, emr-6.0.0, emr-7.0.0
	r5d.4xlarge	emr-5.29.0, emr-6.0.0, emr-7.0.0
	r5d.8xlarge	emr-5.29.0, emr-6.0.0, emr-7.0.0
	r5d.12xlarge	emr-5.29.0, emr-6.0.0, emr-7.0.0
	r5d.16xlarge	emr-5.29.0, emr-6.0.0, emr-7.0.0
	r5d.24xlarge	emr-5.29.0, emr-6.0.0, emr-7.0.0
	r5dn.xlarge	emr-5.31.0, emr-6.1.0, emr-7.0.0
	r5dn.2xlarge	emr-5.31.0, emr-6.1.0, emr-7.0.0
	r5dn.4xlarge	emr-5.31.0, emr-6.1.0, emr-7.0.0
	r5dn.8xlarge	emr-5.31.0, emr-6.1.0, emr-7.0.0
	r5dn.12xlarge	emr-5.31.0, emr-6.1.0, emr-7.0.0
	r5dn.16xlarge	emr-5.31.0, emr-6.1.0, emr-7.0.0

Kelas Instans	Tipe instans	Versi EMR Amazon yang didukung minimum (5.x, 6.x, 7.x)
	r5dn.24xlarge	emr-5.31.0, emr-6.1.0, emr-7.0.0
	r5n.xlarge	emr-5.31.0, emr-6.1.0, emr-7.0.0
	r5n.2xlarge	emr-5.31.0, emr-6.1.0, emr-7.0.0
	r5n.4xlarge	emr-5.31.0, emr-6.1.0, emr-7.0.0
	r5n.8xlarge	emr-5.31.0, emr-6.1.0, emr-7.0.0
	r5n.12xlarge	emr-5.31.0, emr-6.1.0, emr-7.0.0
	r5n.16xlarge	emr-5.31.0, emr-6.1.0, emr-7.0.0
	r5n.24xlarge	emr-5.31.0, emr-6.1.0, emr-7.0.0
	r6g.xlarge	emr-5.31.0, emr-6.1.0, emr-7.0.0
	r6g.2xlarge	emr-5.31.0, emr-6.1.0, emr-7.0.0
	r6g.4xlarge	emr-5.31.0, emr-6.1.0, emr-7.0.0
	r6g.8xlarge	emr-5.31.0, emr-6.1.0, emr-7.0.0

Kelas Instans	Tipe instans	Versi EMR Amazon yang didukung minimum (5.x, 6.x, 7.x)
	r6g.12xlarge	emr-5.31.0, emr-6.1.0, emr-7.0.0
	r6g.16xlarge	emr-5.31.0, emr-6.1.0, emr-7.0.0
	r6i.xlarge	emr-5.35.0, emr-6.6.0
	r6i.2xlarge	emr-5.35.0, emr-6.6.0
	r6i.4xlarge	emr-5.35.0, emr-6.6.0
	r6i.8xlarge	emr-5.35.0, emr-6.6.0
	r6i.12xlarge	emr-5.35.0, emr-6.6.0
	r6i.16xlarge	emr-5.35.0, emr-6.6.0
	r6i.24xlarge	emr-5.35.0, emr-6.6.0
	r6i.32xlarge	emr-5.35.0, emr-6.6.0
	x1.16xlarge	emr-5.36.1, emr-6.9.0, emr-7.0.0
	x1.32xlarge	emr-5.36.1, emr-6.9.0, emr-7.0.0
	x1e.xlarge	emr-5.36.1, emr-6.10.0, emr-7.0.0
	x1e.2xlarge	emr-5.36.1, emr-6.10.0, emr-7.0.0
	x1e.4xlarge	emr-5.36.1, emr-6.10.0, emr-7.0.0

Kelas Instans	Tipe instans	Versi EMR Amazon yang didukung minimum (5.x, 6.x, 7.x)
	x1e.8xlarge	emr-5.36.1, emr-6.10.0, emr-7.0.0
	x1e.16xlarge	emr-5.36.1, emr-6.10.0, emr-7.0.0
	x1e.32xlarge	emr-5.36.1, emr-6.10.0, emr-7.0.0
	x2idn.16xlarge	emr-5.36.1, emr-6.9.0
	x2idn.24xlarge	emr-5.36.1, emr-6.9.0
	x2idn.32xlarge	emr-5.36.1, emr-6.9.0
	x2iedn.xlarge	emr-5.36.1, emr-6.9.0
	x2iedn.2xlarge	emr-5.36.1, emr-6.9.0
	x2iedn.4xlarge	emr-5.36.1, emr-6.9.0
	x2iedn.8xlarge	emr-5.36.1, emr-6.9.0
	x2iedn.16xlarge	emr-5.36.1, emr-6.9.0
	x2iedn.24xlarge	emr-5.36.1, emr-6.9.0
	x2iedn.32xlarge	emr-5.36.1, emr-6.9.0
Penyimpanan Dioptimalkan	i3.xlarge	emr-5.29.0, emr-6.0.0, emr-7.0.0
	i3.2xlarge	emr-5.29.0, emr-6.0.0, emr-7.0.0
	i3.4xlarge	emr-5.29.0, emr-6.0.0, emr-7.0.0

Kelas Instans	Tipe instans	Versi EMR Amazon yang didukung minimum (5.x, 6.x, 7.x)
	i3.8xlarge	emr-5.29.0, emr-6.0.0, emr-7.0.0
	i3.16xlarge	emr-5.29.0, emr-6.0.0, emr-7.0.0
	i3en.xlarge	emr-5.29.0, emr-6.0.0, emr-7.0.0
	i3en.2xlarge	emr-5.29.0, emr-6.0.0, emr-7.0.0
	i3en.3xlarge	emr-5.29.0, emr-6.0.0, emr-7.0.0
	i3en.6xlarge	emr-5.29.0, emr-6.0.0, emr-7.0.0
	i3en.12xlarge	emr-5.29.0, emr-6.0.0, emr-7.0.0
	i3en.24xlarge	emr-5.29.0, emr-6.0.0, emr-7.0.0
	i4i.xlarge	emr-5.36.1, emr-6.8.0
	i4i.2xlarge	emr-5.36.1, emr-6.8.0
	i4i.4xlarge	emr-5.36.1, emr-6.8.0
	i4i.8xlarge	emr-5.36.1, emr-6.8.0
	i4i.12xlarge	emr-5.36.1, emr-6.10.0
	i4i.16xlarge	emr-5.36.1, emr-6.8.0

Kelas Instans	Tipe instans	Versi EMR Amazon yang didukung minimum (5.x, 6.x, 7.x)
	i4i.24xlarge	emr-5.36.1, emr-6.10.0
	i4i.32xlarge	emr-5.36.1, emr-6.8.0

Asia Pasifik (Hong Kong) - ap-timur-1

Kelas Instans	Tipe instans	Versi EMR Amazon yang didukung minimum (5.x, 6.x, 7.x)
Tujuan Umum	m5.xlarge	emr-5.20.0, emr-6.0.0, emr-7.0.0
	m5.2xlarge	emr-5.20.0, emr-6.0.0, emr-7.0.0
	m5.4xlarge	emr-5.20.0, emr-6.0.0, emr-7.0.0
	m5.8xlarge	emr-5.20.0, emr-6.0.0, emr-7.0.0
	m5.12xlarge	emr-5.20.0, emr-6.0.0, emr-7.0.0
	m5.16xlarge	emr-5.20.0, emr-6.0.0, emr-7.0.0
	m5.24xlarge	emr-5.20.0, emr-6.0.0, emr-7.0.0
	m5d.xlarge	emr-5.20.0, emr-6.0.0, emr-7.0.0

Kelas Instans	Tipe instans	Versi EMR Amazon yang didukung minimum (5.x, 6.x, 7.x)
	m5d.2xlarge	emr-5.20.0, emr-6.0.0, emr-7.0.0
	m5d.4xlarge	emr-5.20.0, emr-6.0.0, emr-7.0.0
	m5d.8xlarge	emr-5.20.0, emr-6.0.0, emr-7.0.0
	m5d.12xlarge	emr-5.20.0, emr-6.0.0, emr-7.0.0
	m5d.16xlarge	emr-5.20.0, emr-6.0.0, emr-7.0.0
	m5d.24xlarge	emr-5.20.0, emr-6.0.0, emr-7.0.0
	m6g.xlarge	emr-5.30.0, emr-6.1.0, emr-7.0.0
	m6g.2xlarge	emr-5.30.0, emr-6.1.0, emr-7.0.0
	m6g.4xlarge	emr-5.30.0, emr-6.1.0, emr-7.0.0
	m6g.8xlarge	emr-5.30.0, emr-6.1.0, emr-7.0.0
	m6g.12xlarge	emr-5.30.0, emr-6.1.0, emr-7.0.0
	m6g.16xlarge	emr-5.30.0, emr-6.1.0, emr-7.0.0

Kelas Instans	Tipe instans	Versi EMR Amazon yang didukung minimum (5.x, 6.x, 7.x)
	m6i.xlarge	emr-5.35.0, emr-6.6.0
	m6i.2xlarge	emr-5.35.0, emr-6.6.0
	m6i.4xlarge	emr-5.35.0, emr-6.6.0
	m6i.8xlarge	emr-5.35.0, emr-6.6.0
	m6i.12xlarge	emr-5.35.0, emr-6.6.0
	m6i.16xlarge	emr-5.35.0, emr-6.6.0
	m6i.24xlarge	emr-5.35.0, emr-6.6.0
	m6i.32xlarge	emr-5.35.0, emr-6.6.0
Komputasi Dioptimalkan	c5.xlarge	emr-5.20.0, emr-6.0.0, emr-7.0.0
	c5.2xlarge	emr-5.20.0, emr-6.0.0, emr-7.0.0
	c5.4xlarge	emr-5.20.0, emr-6.0.0, emr-7.0.0
	c5.9xlarge	emr-5.20.0, emr-6.0.0, emr-7.0.0
	c5.12xlarge	emr-5.20.0, emr-6.0.0, emr-7.0.0
	c5.18xlarge	emr-5.20.0, emr-6.0.0, emr-7.0.0
	c5.24xlarge	emr-5.20.0, emr-6.0.0, emr-7.0.0

Kelas Instans	Tipe instans	Versi EMR Amazon yang didukung minimum (5.x, 6.x, 7.x)
	c5a.xlarge	emr-5.31.0, emr-6.1.0, emr-7.0.0
	c5a.2xlarge	emr-5.31.0, emr-6.1.0, emr-7.0.0
	c5a.4xlarge	emr-5.31.0, emr-6.1.0, emr-7.0.0
	c5a.8xlarge	emr-5.31.0, emr-6.1.0, emr-7.0.0
	c5a.12xlarge	emr-5.31.0, emr-6.1.0, emr-7.0.0
	c5a.16xlarge	emr-5.31.0, emr-6.1.0, emr-7.0.0
	c5a.24xlarge	emr-5.31.0, emr-6.1.0, emr-7.0.0
	c5d.xlarge	emr-5.20.0, emr-6.0.0, emr-7.0.0
	c5d.2xlarge	emr-5.20.0, emr-6.0.0, emr-7.0.0
	c5d.4xlarge	emr-5.20.0, emr-6.0.0, emr-7.0.0
	c5d.9xlarge	emr-5.20.0, emr-6.0.0, emr-7.0.0
	c5d.18xlarge	emr-5.20.0, emr-6.0.0, emr-7.0.0

Kelas Instans	Tipe instans	Versi EMR Amazon yang didukung minimum (5.x, 6.x, 7.x)
	c5n.xlarge	emr-5.20.0, emr-6.0.0, emr-7.0.0
	c5n.2xlarge	emr-5.20.0, emr-6.0.0, emr-7.0.0
	c5n.4xlarge	emr-5.20.0, emr-6.0.0, emr-7.0.0
	c5n.9xlarge	emr-5.20.0, emr-6.0.0, emr-7.0.0
	c5n.18xlarge	emr-5.20.0, emr-6.0.0, emr-7.0.0
	c6g.xlarge	emr-5.31.0, emr-6.1.0, emr-7.0.0
	c6g.2xlarge	emr-5.31.0, emr-6.1.0, emr-7.0.0
	c6g.4xlarge	emr-5.31.0, emr-6.1.0, emr-7.0.0
	c6g.8xlarge	emr-5.31.0, emr-6.1.0, emr-7.0.0
	c6g.12xlarge	emr-5.31.0, emr-6.1.0, emr-7.0.0
	c6g.16xlarge	emr-5.31.0, emr-6.1.0, emr-7.0.0
	c6gn.xlarge	emr-5.33.0, emr-6.3.0, emr-7.0.0

Kelas Instans	Tipe instans	Versi EMR Amazon yang didukung minimum (5.x, 6.x, 7.x)
	c6gn.2xlarge	emr-5.33.0, emr-6.3.0, emr-7.0.0
	c6gn.4xlarge	emr-5.33.0, emr-6.3.0, emr-7.0.0
	c6gn.8xlarge	emr-5.33.0, emr-6.3.0, emr-7.0.0
	c6gn.12xlarge	emr-5.33.0, emr-6.3.0, emr-7.0.0
	c6gn.16xlarge	emr-5.33.0, emr-6.3.0, emr-7.0.0
	c6i.xlarge	emr-5.35.0, emr-6.6.0
	c6i.2xlarge	emr-5.35.0, emr-6.6.0
	c6i.4xlarge	emr-5.35.0, emr-6.6.0
	c6i.8xlarge	emr-5.35.0, emr-6.6.0
	c6i.12xlarge	emr-5.35.0, emr-6.6.0
	c6i.16xlarge	emr-5.35.0, emr-6.6.0
	c6i.24xlarge	emr-5.35.0, emr-6.6.0
	c6i.32xlarge	emr-5.35.0, emr-6.6.0
Komputasi yang Dipercepat	g4dn.xlarge	emr-5.30.0, emr-6.0.0, emr-7.0.0
	g4dn.2xlarge	emr-5.30.0, emr-6.0.0, emr-7.0.0

Kelas Instans	Tipe instans	Versi EMR Amazon yang didukung minimum (5.x, 6.x, 7.x)
	g4dn.4xlarge	emr-5.30.0, emr-6.0.0, emr-7.0.0
	g4dn.8xlarge	emr-5.30.0, emr-6.0.0, emr-7.0.0
	g4dn.12xlarge	emr-5.30.0, emr-6.0.0, emr-7.0.0
	g4dn.16xlarge	emr-5.30.0, emr-6.0.0, emr-7.0.0
Memori Dioptimalkan	r5.xlarge	emr-5.20.0, emr-6.0.0, emr-7.0.0
	r5.2xlarge	emr-5.20.0, emr-6.0.0, emr-7.0.0
	r5.4xlarge	emr-5.20.0, emr-6.0.0, emr-7.0.0
	r5.8xlarge	emr-5.20.0, emr-6.0.0, emr-7.0.0
	r5.12xlarge	emr-5.20.0, emr-6.0.0, emr-7.0.0
	r5.16xlarge	emr-5.20.0, emr-6.0.0, emr-7.0.0
	r5.24xlarge	emr-5.20.0, emr-6.0.0, emr-7.0.0
	r5d.xlarge	emr-5.20.0, emr-6.0.0, emr-7.0.0

Kelas Instans	Tipe instans	Versi EMR Amazon yang didukung minimum (5.x, 6.x, 7.x)
	r5d.2xlarge	emr-5.20.0, emr-6.0.0, emr-7.0.0
	r5d.4xlarge	emr-5.20.0, emr-6.0.0, emr-7.0.0
	r5d.8xlarge	emr-5.20.0, emr-6.0.0, emr-7.0.0
	r5d.12xlarge	emr-5.20.0, emr-6.0.0, emr-7.0.0
	r5d.16xlarge	emr-5.20.0, emr-6.0.0, emr-7.0.0
	r5d.24xlarge	emr-5.20.0, emr-6.0.0, emr-7.0.0
	r5n.xlarge	emr-5.31.0, emr-6.1.0, emr-7.0.0
	r5n.2xlarge	emr-5.31.0, emr-6.1.0, emr-7.0.0
	r5n.4xlarge	emr-5.31.0, emr-6.1.0, emr-7.0.0
	r5n.8xlarge	emr-5.31.0, emr-6.1.0, emr-7.0.0
	r5n.12xlarge	emr-5.31.0, emr-6.1.0, emr-7.0.0
	r5n.16xlarge	emr-5.31.0, emr-6.1.0, emr-7.0.0

Kelas Instans	Tipe instans	Versi EMR Amazon yang didukung minimum (5.x, 6.x, 7.x)
	r5n.24xlarge	emr-5.31.0, emr-6.1.0, emr-7.0.0
	r6g.xlarge	emr-5.31.0, emr-6.1.0, emr-7.0.0
	r6g.2xlarge	emr-5.31.0, emr-6.1.0, emr-7.0.0
	r6g.4xlarge	emr-5.31.0, emr-6.1.0, emr-7.0.0
	r6g.8xlarge	emr-5.31.0, emr-6.1.0, emr-7.0.0
	r6g.12xlarge	emr-5.31.0, emr-6.1.0, emr-7.0.0
	r6g.16xlarge	emr-5.31.0, emr-6.1.0, emr-7.0.0
	r6i.xlarge	emr-5.35.0, emr-6.6.0
	r6i.2xlarge	emr-5.35.0, emr-6.6.0
	r6i.4xlarge	emr-5.35.0, emr-6.6.0
	r6i.8xlarge	emr-5.35.0, emr-6.6.0
	r6i.12xlarge	emr-5.35.0, emr-6.6.0
	r6i.16xlarge	emr-5.35.0, emr-6.6.0
	r6i.24xlarge	emr-5.35.0, emr-6.6.0
	r6i.32xlarge	emr-5.35.0, emr-6.6.0

Kelas Instans	Tipe instans	Versi EMR Amazon yang didukung minimum (5.x, 6.x, 7.x)
	x1.16xlarge	emr-5.36.1, emr-6.9.0, emr-7.0.0
	x1.32xlarge	emr-5.36.1, emr-6.9.0, emr-7.0.0
Penyimpanan Dioptimalkan	i3.xlarge	emr-5.20.0, emr-6.0.0, emr-7.0.0
	i3.2xlarge	emr-5.20.0, emr-6.0.0, emr-7.0.0
	i3.4xlarge	emr-5.20.0, emr-6.0.0, emr-7.0.0
	i3.8xlarge	emr-5.20.0, emr-6.0.0, emr-7.0.0
	i3.16xlarge	emr-5.20.0, emr-6.0.0, emr-7.0.0
	i3en.xlarge	emr-5.25.0, emr-6.0.0, emr-7.0.0
	i3en.2xlarge	emr-5.25.0, emr-6.0.0, emr-7.0.0
	i3en.3xlarge	emr-5.25.0, emr-6.0.0, emr-7.0.0
	i3en.6xlarge	emr-5.25.0, emr-6.0.0, emr-7.0.0
	i3en.12xlarge	emr-5.25.0, emr-6.0.0, emr-7.0.0

Kelas Instans	Tipe instans	Versi EMR Amazon yang didukung minimum (5.x, 6.x, 7.x)
	i3en.24xlarge	emr-5.25.0, emr-6.0.0, emr-7.0.0
	i4i.xlarge	emr-5.36.1, emr-6.8.0
	i4i.2xlarge	emr-5.36.1, emr-6.8.0
	i4i.4xlarge	emr-5.36.1, emr-6.8.0
	i4i.8xlarge	emr-5.36.1, emr-6.8.0
	i4i.12xlarge	emr-5.36.1, emr-6.10.0
	i4i.16xlarge	emr-5.36.1, emr-6.8.0
	i4i.24xlarge	emr-5.36.1, emr-6.10.0
	i4i.32xlarge	emr-5.36.1, emr-6.8.0

Asia Pasifik (Jakarta) - ap-tenggara-3

Kelas Instans	Tipe instans	Versi EMR Amazon yang didukung minimum (5.x, 6.x, 7.x)
Tujuan Umum	m5.xlarge	emr-5.30.2, emr-6.0.1, emr-7.0.0
	m5.2xlarge	emr-5.30.2, emr-6.0.1, emr-7.0.0
	m5.4xlarge	emr-5.30.2, emr-6.0.1, emr-7.0.0

Kelas Instans	Tipe instans	Versi EMR Amazon yang didukung minimum (5.x, 6.x, 7.x)
	m5.8xlarge	emr-5.30.2, emr-6.0.1, emr-7.0.0
	m5.12xlarge	emr-5.30.2, emr-6.0.1, emr-7.0.0
	m5.16xlarge	emr-5.30.2, emr-6.0.1, emr-7.0.0
	m5.24xlarge	emr-5.30.2, emr-6.0.1, emr-7.0.0
	m5d.xlarge	emr-5.30.2, emr-6.0.1, emr-7.0.0
	m5d.2xlarge	emr-5.30.2, emr-6.0.1, emr-7.0.0
	m5d.4xlarge	emr-5.30.2, emr-6.0.1, emr-7.0.0
	m5d.8xlarge	emr-5.30.2, emr-6.0.1, emr-7.0.0
	m5d.12xlarge	emr-5.30.2, emr-6.0.1, emr-7.0.0
	m5d.16xlarge	emr-5.30.2, emr-6.0.1, emr-7.0.0
	m5d.24xlarge	emr-5.30.2, emr-6.0.1, emr-7.0.0
	m6g.xlarge	emr-5.30.2, emr-6.1.1, emr-7.0.0

Kelas Instans	Tipe instans	Versi EMR Amazon yang didukung minimum (5.x, 6.x, 7.x)
	m6g.2xlarge	emr-5.30.2, emr-6.1.1, emr-7.0.0
	m6g.4xlarge	emr-5.30.2, emr-6.1.1, emr-7.0.0
	m6g.8xlarge	emr-5.30.2, emr-6.1.1, emr-7.0.0
	m6g.12xlarge	emr-5.30.2, emr-6.1.1, emr-7.0.0
	m6g.16xlarge	emr-5.30.2, emr-6.1.1, emr-7.0.0
	m6gd.xlarge	emr-5.33.0, emr-6.3.0, emr-7.0.0
	m6gd.2xlarge	emr-5.33.0, emr-6.3.0, emr-7.0.0
	m6gd.4xlarge	emr-5.33.0, emr-6.3.0, emr-7.0.0
	m6gd.8xlarge	emr-5.33.0, emr-6.3.0, emr-7.0.0
	m6gd.12xlarge	emr-5.33.0, emr-6.3.0, emr-7.0.0
	m6gd.16xlarge	emr-5.33.0, emr-6.3.0, emr-7.0.0
	m6i.xlarge	emr-5.35.0, emr-6.6.0
	m6i.2xlarge	emr-5.35.0, emr-6.6.0

Kelas Instans	Tipe instans	Versi EMR Amazon yang didukung minimum (5.x, 6.x, 7.x)
	m6i.4xlarge	emr-5.35.0, emr-6.6.0
	m6i.8xlarge	emr-5.35.0, emr-6.6.0
	m6i.12xlarge	emr-5.35.0, emr-6.6.0
	m6i.16xlarge	emr-5.35.0, emr-6.6.0
	m6i.24xlarge	emr-5.35.0, emr-6.6.0
	m6i.32xlarge	emr-5.35.0, emr-6.6.0
Komputasi Dioptimalkan	c5.xlarge	emr-5.30.2, emr-6.0.1, emr-7.0.0
	c5.2xlarge	emr-5.30.2, emr-6.0.1, emr-7.0.0
	c5.4xlarge	emr-5.30.2, emr-6.0.1, emr-7.0.0
	c5.9xlarge	emr-5.30.2, emr-6.0.1, emr-7.0.0
	c5.12xlarge	emr-5.30.2, emr-6.0.1, emr-7.0.0
	c5.18xlarge	emr-5.30.2, emr-6.0.1, emr-7.0.0
	c5.24xlarge	emr-5.30.2, emr-6.0.1, emr-7.0.0
	c5d.xlarge	emr-5.30.2, emr-6.0.1, emr-7.0.0

Kelas Instans	Tipe instans	Versi EMR Amazon yang didukung minimum (5.x, 6.x, 7.x)
	c5d.2xlarge	emr-5.30.2, emr-6.0.1, emr-7.0.0
	c5d.4xlarge	emr-5.30.2, emr-6.0.1, emr-7.0.0
	c5d.9xlarge	emr-5.30.2, emr-6.0.1, emr-7.0.0
	c5d.12xlarge	emr-5.30.2, emr-6.0.1, emr-7.0.0
	c5d.18xlarge	emr-5.30.2, emr-6.0.1, emr-7.0.0
	c5d.24xlarge	emr-5.30.2, emr-6.0.1, emr-7.0.0
	c5n.xlarge	emr-5.30.2, emr-6.0.1, emr-7.0.0
	c5n.2xlarge	emr-5.30.2, emr-6.0.1, emr-7.0.0
	c5n.4xlarge	emr-5.30.2, emr-6.0.1, emr-7.0.0
	c5n.9xlarge	emr-5.30.2, emr-6.0.1, emr-7.0.0
	c5n.18xlarge	emr-5.30.2, emr-6.0.1, emr-7.0.0
	c6g.xlarge	emr-5.31.1, emr-6.1.1, emr-7.0.0

Kelas Instans	Tipe instans	Versi EMR Amazon yang didukung minimum (5.x, 6.x, 7.x)
	c6g.2xlarge	emr-5.31.1, emr-6.1.1, emr-7.0.0
	c6g.4xlarge	emr-5.31.1, emr-6.1.1, emr-7.0.0
	c6g.8xlarge	emr-5.31.1, emr-6.1.1, emr-7.0.0
	c6g.12xlarge	emr-5.31.1, emr-6.1.1, emr-7.0.0
	c6g.16xlarge	emr-5.31.1, emr-6.1.1, emr-7.0.0
	c6gd.xlarge	emr-5.33.0, emr-6.3.0, emr-7.0.0
	c6gd.2xlarge	emr-5.33.0, emr-6.3.0, emr-7.0.0
	c6gd.4xlarge	emr-5.33.0, emr-6.3.0, emr-7.0.0
	c6gd.8xlarge	emr-5.33.0, emr-6.3.0, emr-7.0.0
	c6gd.12xlarge	emr-5.33.0, emr-6.3.0, emr-7.0.0
	c6gd.16xlarge	emr-5.33.0, emr-6.3.0, emr-7.0.0
	c6in.xlarge	emr-5.36.1, emr-6.10.0
	c6in.2xlarge	emr-5.36.1, emr-6.10.0

Kelas Instans	Tipe instans	Versi EMR Amazon yang didukung minimum (5.x, 6.x, 7.x)
	c6in.4xlarge	emr-5.36.1, emr-6.10.0
	c6in.8xlarge	emr-5.36.1, emr-6.10.0
	c6in.12xlarge	emr-5.36.1, emr-6.10.0
	c6in.16xlarge	emr-5.36.1, emr-6.10.0
	c6in.24xlarge	emr-5.36.1, emr-6.10.0
	c6in.32xlarge	emr-5.36.1, emr-6.10.0
Komputasi yang Dipercepat	g5.xlarge	emr-5.36.1, emr-6.9.0, emr-7.0.0
	g5.2xlarge	emr-5.36.1, emr-6.9.0, emr-7.0.0
	g5.4xlarge	emr-5.36.1, emr-6.9.0, emr-7.0.0
	g5.8xlarge	emr-5.36.1, emr-6.9.0, emr-7.0.0
	g5.12xlarge	emr-5.36.1, emr-6.9.0, emr-7.0.0
	g5.16xlarge	emr-5.36.1, emr-6.9.0, emr-7.0.0
	g5.24xlarge	emr-5.36.1, emr-6.9.0, emr-7.0.0
	g5.48xlarge	emr-5.36.1, emr-6.9.0, emr-7.0.0

Kelas Instans	Tipe instans	Versi EMR Amazon yang didukung minimum (5.x, 6.x, 7.x)
Memori Dioptimalkan	r5.xlarge	emr-5.30.2, emr-6.0.1, emr-7.0.0
	r5.2xlarge	emr-5.30.2, emr-6.0.1, emr-7.0.0
	r5.4xlarge	emr-5.30.2, emr-6.0.1, emr-7.0.0
	r5.8xlarge	emr-5.30.2, emr-6.0.1, emr-7.0.0
	r5.12xlarge	emr-5.30.2, emr-6.0.1, emr-7.0.0
	r5.16xlarge	emr-5.30.2, emr-6.0.1, emr-7.0.0
	r5.24xlarge	emr-5.30.2, emr-6.0.1, emr-7.0.0
	r5d.xlarge	emr-5.30.2, emr-6.0.1, emr-7.0.0
	r5d.2xlarge	emr-5.30.2, emr-6.0.1, emr-7.0.0
	r5d.4xlarge	emr-5.30.2, emr-6.0.1, emr-7.0.0
	r5d.8xlarge	emr-5.30.2, emr-6.0.1, emr-7.0.0
	r5d.12xlarge	emr-5.30.2, emr-6.0.1, emr-7.0.0

Kelas Instans	Tipe instans	Versi EMR Amazon yang didukung minimum (5.x, 6.x, 7.x)
	r5d.16xlarge	emr-5.30.2, emr-6.0.1, emr-7.0.0
	r5d.24xlarge	emr-5.30.2, emr-6.0.1, emr-7.0.0
	r6g.xlarge	emr-5.31.1, emr-6.1.1, emr-7.0.0
	r6g.2xlarge	emr-5.31.1, emr-6.1.1, emr-7.0.0
	r6g.4xlarge	emr-5.31.1, emr-6.1.1, emr-7.0.0
	r6g.8xlarge	emr-5.31.1, emr-6.1.1, emr-7.0.0
	r6g.12xlarge	emr-5.31.1, emr-6.1.1, emr-7.0.0
	r6g.16xlarge	emr-5.31.1, emr-6.1.1, emr-7.0.0
	r6gd.xlarge	emr-5.33.0, emr-6.3.0, emr-7.0.0
	r6gd.2xlarge	emr-5.33.0, emr-6.3.0, emr-7.0.0
	r6gd.4xlarge	emr-5.33.0, emr-6.3.0, emr-7.0.0
	r6gd.8xlarge	emr-5.33.0, emr-6.3.0, emr-7.0.0

Kelas Instans	Tipe instans	Versi EMR Amazon yang didukung minimum (5.x, 6.x, 7.x)
	r6gd.12xlarge	emr-5.33.0, emr-6.3.0, emr-7.0.0
	r6gd.16xlarge	emr-5.33.0, emr-6.3.0, emr-7.0.0
	x2idn.16xlarge	emr-5.36.1, emr-6.9.0
	x2idn.24xlarge	emr-5.36.1, emr-6.9.0
	x2idn.32xlarge	emr-5.36.1, emr-6.9.0
	x2iedn.xlarge	emr-5.36.1, emr-6.9.0
	x2iedn.2xlarge	emr-5.36.1, emr-6.9.0
	x2iedn.4xlarge	emr-5.36.1, emr-6.9.0
	x2iedn.8xlarge	emr-5.36.1, emr-6.9.0
	x2iedn.16xlarge	emr-5.36.1, emr-6.9.0
	x2iedn.24xlarge	emr-5.36.1, emr-6.9.0
	x2iedn.32xlarge	emr-5.36.1, emr-6.9.0
Penyimpanan Dioptimalkan	i3.xlarge	emr-5.30.2, emr-6.0.1, emr-7.0.0
	i3.2xlarge	emr-5.30.2, emr-6.0.1, emr-7.0.0
	i3.4xlarge	emr-5.30.2, emr-6.0.1, emr-7.0.0
	i3.8xlarge	emr-5.30.2, emr-6.0.1, emr-7.0.0

Kelas Instans	Tipe instans	Versi EMR Amazon yang didukung minimum (5.x, 6.x, 7.x)
	i3.16xlarge	emr-5.30.2, emr-6.0.1, emr-7.0.0
	i3en.xlarge	emr-5.30.2, emr-6.0.1, emr-7.0.0
	i3en.2xlarge	emr-5.30.2, emr-6.0.1, emr-7.0.0
	i3en.3xlarge	emr-5.30.2, emr-6.0.1, emr-7.0.0
	i3en.6xlarge	emr-5.30.2, emr-6.0.1, emr-7.0.0
	i3en.12xlarge	emr-5.30.2, emr-6.0.1, emr-7.0.0
	i3en.24xlarge	emr-5.30.2, emr-6.0.1, emr-7.0.0
	i4i.xlarge	emr-5.36.1, emr-6.8.0
	i4i.2xlarge	emr-5.36.1, emr-6.8.0
	i4i.4xlarge	emr-5.36.1, emr-6.8.0
	i4i.8xlarge	emr-5.36.1, emr-6.8.0
	i4i.12xlarge	emr-5.36.1, emr-6.10.0
	i4i.16xlarge	emr-5.36.1, emr-6.8.0
	i4i.24xlarge	emr-5.36.1, emr-6.10.0
	i4i.32xlarge	emr-5.36.1, emr-6.8.0

Asia Pacific (Mumbai) - ap-south-1

Kelas Instans	Tipe instans	Versi EMR Amazon yang didukung minimum (5.x, 6.x, 7.x)
Tujuan Umum	m5.xlarge	emr-5.13.0, emr-6.0.0, emr-7.0.0
	m5.2xlarge	emr-5.13.0, emr-6.0.0, emr-7.0.0
	m5.4xlarge	emr-5.13.0, emr-6.0.0, emr-7.0.0
	m5.8xlarge	emr-5.13.0, emr-6.0.0, emr-7.0.0
	m5.12xlarge	emr-5.13.0, emr-6.0.0, emr-7.0.0
	m5.16xlarge	emr-5.13.0, emr-6.0.0, emr-7.0.0
	m5.24xlarge	emr-5.13.0, emr-6.0.0, emr-7.0.0
	m5a.xlarge	emr-5.20.0, emr-6.0.0, emr-7.0.0
	m5a.2xlarge	emr-5.20.0, emr-6.0.0, emr-7.0.0
	m5a.4xlarge	emr-5.20.0, emr-6.0.0, emr-7.0.0
	m5a.8xlarge	emr-5.20.0, emr-6.0.0, emr-7.0.0

Kelas Instans	Tipe instans	Versi EMR Amazon yang didukung minimum (5.x, 6.x, 7.x)
	m5a.12xlarge	emr-5.20.0, emr-6.0.0, emr-7.0.0
	m5a.16xlarge	emr-5.20.0, emr-6.0.0, emr-7.0.0
	m5a.24xlarge	emr-5.20.0, emr-6.0.0, emr-7.0.0
	m5ad.xlarge	emr-5.20.0, emr-6.0.0, emr-7.0.0
	m5ad.2xlarge	emr-5.20.0, emr-6.0.0, emr-7.0.0
	m5ad.4xlarge	emr-5.20.0, emr-6.0.0, emr-7.0.0
	m5ad.8xlarge	emr-5.33.0, emr-6.3.0, emr-7.0.0
	m5ad.12xlarge	emr-5.20.0, emr-6.0.0, emr-7.0.0
	m5ad.16xlarge	emr-5.33.0, emr-6.3.0, emr-7.0.0
	m5ad.24xlarge	emr-5.20.0, emr-6.0.0, emr-7.0.0
	m5d.xlarge	emr-5.13.0, emr-6.0.0, emr-7.0.0
	m5d.2xlarge	emr-5.13.0, emr-6.0.0, emr-7.0.0

Kelas Instans	Tipe instans	Versi EMR Amazon yang didukung minimum (5.x, 6.x, 7.x)
	m5d.4xlarge	emr-5.13.0, emr-6.0.0, emr-7.0.0
	m5d.8xlarge	emr-5.13.0, emr-6.0.0, emr-7.0.0
	m5d.12xlarge	emr-5.13.0, emr-6.0.0, emr-7.0.0
	m5d.16xlarge	emr-5.13.0, emr-6.0.0, emr-7.0.0
	m5d.24xlarge	emr-5.13.0, emr-6.0.0, emr-7.0.0
	m6a.xlarge	emr-5.36.1, emr-6.8.0, emr-7.0.0
	m6a.2xlarge	emr-5.36.1, emr-6.8.0, emr-7.0.0
	m6a.4xlarge	emr-5.36.1, emr-6.8.0, emr-7.0.0
	m6a.8xlarge	emr-5.36.1, emr-6.8.0, emr-7.0.0
	m6a.12xlarge	emr-5.36.1, emr-6.8.0, emr-7.0.0
	m6a.16xlarge	emr-5.36.1, emr-6.8.0, emr-7.0.0
	m6a.24xlarge	emr-5.36.1, emr-6.8.0, emr-7.0.0

Kelas Instans	Tipe instans	Versi EMR Amazon yang didukung minimum (5.x, 6.x, 7.x)
	m6a.32xlarge	emr-5.36.1, emr-6.8.0, emr-7.0.0
	m6a.48xlarge	emr-5.36.1, emr-6.8.0, emr-7.0.0
	m6g.xlarge	emr-5.30.0, emr-6.1.0, emr-7.0.0
	m6g.2xlarge	emr-5.30.0, emr-6.1.0, emr-7.0.0
	m6g.4xlarge	emr-5.30.0, emr-6.1.0, emr-7.0.0
	m6g.8xlarge	emr-5.30.0, emr-6.1.0, emr-7.0.0
	m6g.12xlarge	emr-5.30.0, emr-6.1.0, emr-7.0.0
	m6g.16xlarge	emr-5.30.0, emr-6.1.0, emr-7.0.0
	m6gd.xlarge	emr-5.33.0, emr-6.3.0, emr-7.0.0
	m6gd.2xlarge	emr-5.33.0, emr-6.3.0, emr-7.0.0
	m6gd.4xlarge	emr-5.33.0, emr-6.3.0, emr-7.0.0
	m6gd.8xlarge	emr-5.33.0, emr-6.3.0, emr-7.0.0

Kelas Instans	Tipe instans	Versi EMR Amazon yang didukung minimum (5.x, 6.x, 7.x)
	m6gd.12xlarge	emr-5.33.0, emr-6.3.0, emr-7.0.0
	m6gd.16xlarge	emr-5.33.0, emr-6.3.0, emr-7.0.0
	m6i.xlarge	emr-5.35.0, emr-6.6.0
	m6i.2xlarge	emr-5.35.0, emr-6.6.0
	m6i.4xlarge	emr-5.35.0, emr-6.6.0
	m6i.8xlarge	emr-5.35.0, emr-6.6.0
	m6i.12xlarge	emr-5.35.0, emr-6.6.0
	m6i.16xlarge	emr-5.35.0, emr-6.6.0
	m6i.24xlarge	emr-5.35.0, emr-6.6.0
	m6i.32xlarge	emr-5.35.0, emr-6.6.0
	m7g.xlarge	emr-5.36.1, emr-6.10.0, emr-7.0.0
	m7g.2xlarge	emr-5.36.1, emr-6.10.0, emr-7.0.0
	m7g.4xlarge	emr-5.36.1, emr-6.10.0, emr-7.0.0
	m7g.8xlarge	emr-5.36.1, emr-6.10.0, emr-7.0.0
	m7g.12xlarge	emr-5.36.1, emr-6.10.0, emr-7.0.0

Kelas Instans	Tipe instans	Versi EMR Amazon yang didukung minimum (5.x, 6.x, 7.x)
	m7g.16xlarge	emr-5.36.1, emr-6.10.0, emr-7.0.0
	m7gd.xlarge	emr-5.36.1, emr-6.10.0, emr-7.0.0
	m7gd.2xlarge	emr-5.36.1, emr-6.10.0, emr-7.0.0
	m7gd.4xlarge	emr-5.36.1, emr-6.10.0, emr-7.0.0
	m7gd.8xlarge	emr-5.36.1, emr-6.10.0, emr-7.0.0
	m7gd.12xlarge	emr-5.36.1, emr-6.10.0, emr-7.0.0
	m7gd.16xlarge	emr-5.36.1, emr-6.10.0, emr-7.0.0
Komputasi Dioptimalkan	c5.xlarge	emr-5.13.0, emr-6.0.0, emr-7.0.0
	c5.2xlarge	emr-5.13.0, emr-6.0.0, emr-7.0.0
	c5.4xlarge	emr-5.13.0, emr-6.0.0, emr-7.0.0
	c5.9xlarge	emr-5.13.0, emr-6.0.0, emr-7.0.0
	c5.12xlarge	emr-5.13.0, emr-6.0.0, emr-7.0.0

Kelas Instans	Tipe instans	Versi EMR Amazon yang didukung minimum (5.x, 6.x, 7.x)
	c5.18xlarge	emr-5.13.0, emr-6.0.0, emr-7.0.0
	c5.24xlarge	emr-5.13.0, emr-6.0.0, emr-7.0.0
	c5a.xlarge	emr-5.31.0, emr-6.1.0, emr-7.0.0
	c5a.2xlarge	emr-5.31.0, emr-6.1.0, emr-7.0.0
	c5a.4xlarge	emr-5.31.0, emr-6.1.0, emr-7.0.0
	c5a.8xlarge	emr-5.31.0, emr-6.1.0, emr-7.0.0
	c5a.12xlarge	emr-5.31.0, emr-6.1.0, emr-7.0.0
	c5a.16xlarge	emr-5.31.0, emr-6.1.0, emr-7.0.0
	c5a.24xlarge	emr-5.31.0, emr-6.1.0, emr-7.0.0
	c5d.xlarge	emr-5.13.0, emr-6.0.0, emr-7.0.0
	c5d.2xlarge	emr-5.13.0, emr-6.0.0, emr-7.0.0
	c5d.4xlarge	emr-5.13.0, emr-6.0.0, emr-7.0.0

Kelas Instans	Tipe instans	Versi EMR Amazon yang didukung minimum (5.x, 6.x, 7.x)
	c5d.9xlarge	emr-5.13.0, emr-6.0.0, emr-7.0.0
	c5d.12xlarge	emr-5.13.0, emr-6.0.0, emr-7.0.0
	c5d.18xlarge	emr-5.13.0, emr-6.0.0, emr-7.0.0
	c5d.24xlarge	emr-5.13.0, emr-6.0.0, emr-7.0.0
	c5n.xlarge	emr-5.20.0, emr-6.0.0, emr-7.0.0
	c5n.2xlarge	emr-5.20.0, emr-6.0.0, emr-7.0.0
	c5n.4xlarge	emr-5.20.0, emr-6.0.0, emr-7.0.0
	c5n.9xlarge	emr-5.20.0, emr-6.0.0, emr-7.0.0
	c5n.18xlarge	emr-5.20.0, emr-6.0.0, emr-7.0.0
	c6a.xlarge	emr-5.36.1, emr-6.8.0, emr-7.0.0
	c6a.2xlarge	emr-5.36.1, emr-6.8.0, emr-7.0.0
	c6a.4xlarge	emr-5.36.1, emr-6.8.0, emr-7.0.0

Kelas Instans	Tipe instans	Versi EMR Amazon yang didukung minimum (5.x, 6.x, 7.x)
	c6a.8xlarge	emr-5.36.1, emr-6.8.0, emr-7.0.0
	c6a.12xlarge	emr-5.36.1, emr-6.8.0, emr-7.0.0
	c6a.16xlarge	emr-5.36.1, emr-6.8.0, emr-7.0.0
	c6a.24xlarge	emr-5.36.1, emr-6.8.0, emr-7.0.0
	c6a.32xlarge	emr-5.36.1, emr-6.8.0, emr-7.0.0
	c6a.48xlarge	emr-5.36.1, emr-6.8.0, emr-7.0.0
	c6g.xlarge	emr-5.31.0, emr-6.1.0, emr-7.0.0
	c6g.2xlarge	emr-5.31.0, emr-6.1.0, emr-7.0.0
	c6g.4xlarge	emr-5.31.0, emr-6.1.0, emr-7.0.0
	c6g.8xlarge	emr-5.31.0, emr-6.1.0, emr-7.0.0
	c6g.12xlarge	emr-5.31.0, emr-6.1.0, emr-7.0.0
	c6g.16xlarge	emr-5.31.0, emr-6.1.0, emr-7.0.0

Kelas Instans	Tipe instans	Versi EMR Amazon yang didukung minimum (5.x, 6.x, 7.x)
	c6gd.xlarge	emr-5.33.0, emr-6.3.0, emr-7.0.0
	c6gd.2xlarge	emr-5.33.0, emr-6.3.0, emr-7.0.0
	c6gd.4xlarge	emr-5.33.0, emr-6.3.0, emr-7.0.0
	c6gd.8xlarge	emr-5.33.0, emr-6.3.0, emr-7.0.0
	c6gd.12xlarge	emr-5.33.0, emr-6.3.0, emr-7.0.0
	c6gd.16xlarge	emr-5.33.0, emr-6.3.0, emr-7.0.0
	c6gn.xlarge	emr-5.33.0, emr-6.3.0, emr-7.0.0
	c6gn.2xlarge	emr-5.33.0, emr-6.3.0, emr-7.0.0
	c6gn.4xlarge	emr-5.33.0, emr-6.3.0, emr-7.0.0
	c6gn.8xlarge	emr-5.33.0, emr-6.3.0, emr-7.0.0
	c6gn.12xlarge	emr-5.33.0, emr-6.3.0, emr-7.0.0
	c6gn.16xlarge	emr-5.33.0, emr-6.3.0, emr-7.0.0

Kelas Instans	Tipe instans	Versi EMR Amazon yang didukung minimum (5.x, 6.x, 7.x)
	c6i.xlarge	emr-5.35.0, emr-6.6.0
	c6i.2xlarge	emr-5.35.0, emr-6.6.0
	c6i.4xlarge	emr-5.35.0, emr-6.6.0
	c6i.8xlarge	emr-5.35.0, emr-6.6.0
	c6i.12xlarge	emr-5.35.0, emr-6.6.0
	c6i.16xlarge	emr-5.35.0, emr-6.6.0
	c6i.24xlarge	emr-5.35.0, emr-6.6.0
	c6i.32xlarge	emr-5.35.0, emr-6.6.0
	c6in.xlarge	emr-5.36.1, emr-6.10.0
	c6in.2xlarge	emr-5.36.1, emr-6.10.0
	c6in.4xlarge	emr-5.36.1, emr-6.10.0
	c6in.8xlarge	emr-5.36.1, emr-6.10.0
	c6in.12xlarge	emr-5.36.1, emr-6.10.0
	c6in.16xlarge	emr-5.36.1, emr-6.10.0
	c6in.24xlarge	emr-5.36.1, emr-6.10.0
	c6in.32xlarge	emr-5.36.1, emr-6.10.0
	c7g.xlarge	emr-5.36.1, emr-6.7.0, emr-7.0.0
	c7g.2xlarge	emr-5.36.1, emr-6.7.0, emr-7.0.0

Kelas Instans	Tipe instans	Versi EMR Amazon yang didukung minimum (5.x, 6.x, 7.x)
	c7g.4xlarge	emr-5.36.1, emr-6.7.0, emr-7.0.0
	c7g.8xlarge	emr-5.36.1, emr-6.7.0, emr-7.0.0
	c7g.12xlarge	emr-5.36.1, emr-6.7.0, emr-7.0.0
	c7g.16xlarge	emr-5.36.1, emr-6.7.0, emr-7.0.0
	c7gd.xlarge	emr-5.36.1, emr-6.10.0, emr-7.0.0
	c7gd.2xlarge	emr-5.36.1, emr-6.10.0, emr-7.0.0
	c7gd.4xlarge	emr-5.36.1, emr-6.10.0, emr-7.0.0
	c7gd.8xlarge	emr-5.36.1, emr-6.10.0, emr-7.0.0
	c7gd.12xlarge	emr-5.36.1, emr-6.10.0, emr-7.0.0
	c7gd.16xlarge	emr-5.36.1, emr-6.10.0, emr-7.0.0
Komputasi yang Dipercepat	g4dn.xlarge	emr-5.30.0, emr-6.0.0, emr-7.0.0
	g4dn.2xlarge	emr-5.30.0, emr-6.0.0, emr-7.0.0

Kelas Instans	Tipe instans	Versi EMR Amazon yang didukung minimum (5.x, 6.x, 7.x)
	g4dn.4xlarge	emr-5.30.0, emr-6.0.0, emr-7.0.0
	g4dn.8xlarge	emr-5.30.0, emr-6.0.0, emr-7.0.0
	g4dn.12xlarge	emr-5.30.0, emr-6.0.0, emr-7.0.0
	g4dn.16xlarge	emr-5.30.0, emr-6.0.0, emr-7.0.0
	g5.xlarge	emr-5.36.1, emr-6.9.0, emr-7.0.0
	g5.2xlarge	emr-5.36.1, emr-6.9.0, emr-7.0.0
	g5.4xlarge	emr-5.36.1, emr-6.9.0, emr-7.0.0
	g5.8xlarge	emr-5.36.1, emr-6.9.0, emr-7.0.0
	g5.12xlarge	emr-5.36.1, emr-6.9.0, emr-7.0.0
	g5.16xlarge	emr-5.36.1, emr-6.9.0, emr-7.0.0
	g5.24xlarge	emr-5.36.1, emr-6.9.0, emr-7.0.0
	g5.48xlarge	emr-5.36.1, emr-6.9.0, emr-7.0.0

Kelas Instans	Tipe instans	Versi EMR Amazon yang didukung minimum (5.x, 6.x, 7.x)
	p2.xlarge	emr-5.10.0, emr-6.0.0, emr-7.0.0
	p2.8xlarge	emr-5.10.0, emr-6.0.0, emr-7.0.0
	p2.16xlarge	emr-5.10.0, emr-6.0.0, emr-7.0.0
Memori Dioptimalkan	r5.xlarge	emr-5.13.0, emr-6.0.0, emr-7.0.0
	r5.2xlarge	emr-5.13.0, emr-6.0.0, emr-7.0.0
	r5.4xlarge	emr-5.13.0, emr-6.0.0, emr-7.0.0
	r5.8xlarge	emr-5.13.0, emr-6.0.0, emr-7.0.0
	r5.12xlarge	emr-5.13.0, emr-6.0.0, emr-7.0.0
	r5.16xlarge	emr-5.13.0, emr-6.0.0, emr-7.0.0
	r5.24xlarge	emr-5.13.0, emr-6.0.0, emr-7.0.0
	r5a.xlarge	emr-5.20.0, emr-6.0.0, emr-7.0.0
	r5a.2xlarge	emr-5.20.0, emr-6.0.0, emr-7.0.0

Kelas Instans	Tipe instans	Versi EMR Amazon yang didukung minimum (5.x, 6.x, 7.x)
	r5a.4xlarge	emr-5.20.0, emr-6.0.0, emr-7.0.0
	r5a.8xlarge	emr-5.20.0, emr-6.0.0, emr-7.0.0
	r5a.12xlarge	emr-5.20.0, emr-6.0.0, emr-7.0.0
	r5a.16xlarge	emr-5.20.0, emr-6.0.0, emr-7.0.0
	r5a.24xlarge	emr-5.20.0, emr-6.0.0, emr-7.0.0
	r5ad.xlarge	emr-5.20.0, emr-6.0.0, emr-7.0.0
	r5ad.2xlarge	emr-5.20.0, emr-6.0.0, emr-7.0.0
	r5ad.4xlarge	emr-5.20.0, emr-6.0.0, emr-7.0.0
	r5ad.8xlarge	emr-5.33.0, emr-6.3.0, emr-7.0.0
	r5ad.12xlarge	emr-5.20.0, emr-6.0.0, emr-7.0.0
	r5ad.16xlarge	emr-5.33.0, emr-6.3.0, emr-7.0.0
	r5ad.24xlarge	emr-5.20.0, emr-6.0.0, emr-7.0.0

Kelas Instans	Tipe instans	Versi EMR Amazon yang didukung minimum (5.x, 6.x, 7.x)
	r5d.xlarge	emr-5.13.0, emr-6.0.0, emr-7.0.0
	r5d.2xlarge	emr-5.13.0, emr-6.0.0, emr-7.0.0
	r5d.4xlarge	emr-5.13.0, emr-6.0.0, emr-7.0.0
	r5d.8xlarge	emr-5.13.0, emr-6.0.0, emr-7.0.0
	r5d.12xlarge	emr-5.13.0, emr-6.0.0, emr-7.0.0
	r5d.16xlarge	emr-5.13.0, emr-6.0.0, emr-7.0.0
	r5d.24xlarge	emr-5.13.0, emr-6.0.0, emr-7.0.0
	r5n.xlarge	emr-5.31.0, emr-6.1.0, emr-7.0.0
	r5n.2xlarge	emr-5.31.0, emr-6.1.0, emr-7.0.0
	r5n.4xlarge	emr-5.31.0, emr-6.1.0, emr-7.0.0
	r5n.8xlarge	emr-5.31.0, emr-6.1.0, emr-7.0.0
	r5n.12xlarge	emr-5.31.0, emr-6.1.0, emr-7.0.0

Kelas Instans	Tipe instans	Versi EMR Amazon yang didukung minimum (5.x, 6.x, 7.x)
	r5n.16xlarge	emr-5.31.0, emr-6.1.0, emr-7.0.0
	r5n.24xlarge	emr-5.31.0, emr-6.1.0, emr-7.0.0
	r6a.xlarge	emr-5.36.1, emr-6.8.0, emr-7.0.0
	r6a.2xlarge	emr-5.36.1, emr-6.8.0, emr-7.0.0
	r6a.4xlarge	emr-5.36.1, emr-6.8.0, emr-7.0.0
	r6a.8xlarge	emr-5.36.1, emr-6.8.0, emr-7.0.0
	r6a.12xlarge	emr-5.36.1, emr-6.8.0, emr-7.0.0
	r6a.16xlarge	emr-5.36.1, emr-6.8.0, emr-7.0.0
	r6a.24xlarge	emr-5.36.1, emr-6.8.0, emr-7.0.0
	r6a.32xlarge	emr-5.36.1, emr-6.8.0, emr-7.0.0
	r6a.48xlarge	emr-5.36.1, emr-6.8.0, emr-7.0.0
	r6g.xlarge	emr-5.31.0, emr-6.1.0, emr-7.0.0

Kelas Instans	Tipe instans	Versi EMR Amazon yang didukung minimum (5.x, 6.x, 7.x)
	r6g.2xlarge	emr-5.31.0, emr-6.1.0, emr-7.0.0
	r6g.4xlarge	emr-5.31.0, emr-6.1.0, emr-7.0.0
	r6g.8xlarge	emr-5.31.0, emr-6.1.0, emr-7.0.0
	r6g.12xlarge	emr-5.31.0, emr-6.1.0, emr-7.0.0
	r6g.16xlarge	emr-5.31.0, emr-6.1.0, emr-7.0.0
	r6gd.xlarge	emr-5.33.0, emr-6.3.0, emr-7.0.0
	r6gd.2xlarge	emr-5.33.0, emr-6.3.0, emr-7.0.0
	r6gd.4xlarge	emr-5.33.0, emr-6.3.0, emr-7.0.0
	r6gd.8xlarge	emr-5.33.0, emr-6.3.0, emr-7.0.0
	r6gd.12xlarge	emr-5.33.0, emr-6.3.0, emr-7.0.0
	r6gd.16xlarge	emr-5.33.0, emr-6.3.0, emr-7.0.0
	r6i.xlarge	emr-5.35.0, emr-6.6.0
	r6i.2xlarge	emr-5.35.0, emr-6.6.0

Kelas Instans	Tipe instans	Versi EMR Amazon yang didukung minimum (5.x, 6.x, 7.x)
	r6i.4xlarge	emr-5.35.0, emr-6.6.0
	r6i.8xlarge	emr-5.35.0, emr-6.6.0
	r6i.12xlarge	emr-5.35.0, emr-6.6.0
	r6i.16xlarge	emr-5.35.0, emr-6.6.0
	r6i.24xlarge	emr-5.35.0, emr-6.6.0
	r6i.32xlarge	emr-5.35.0, emr-6.6.0
	r6id.xlarge	emr-5.36.1, emr-6.8.0
	r6id.2xlarge	emr-5.36.1, emr-6.8.0
	r6id.4xlarge	emr-5.36.1, emr-6.8.0
	r6id.8xlarge	emr-5.36.1, emr-6.8.0
	r6id.12xlarge	emr-5.36.1, emr-6.8.0
	r6id.16xlarge	emr-5.36.1, emr-6.8.0
	r6id.24xlarge	emr-5.36.1, emr-6.8.0
	r6id.32xlarge	emr-5.36.1, emr-6.8.0
	r7g.xlarge	emr-5.36.1, emr-6.10.0, emr-7.0.0
	r7g.2xlarge	emr-5.36.1, emr-6.10.0, emr-7.0.0
	r7g.4xlarge	emr-5.36.1, emr-6.10.0, emr-7.0.0

Kelas Instans	Tipe instans	Versi EMR Amazon yang didukung minimum (5.x, 6.x, 7.x)
	r7g.8xlarge	emr-5.36.1, emr-6.10.0, emr-7.0.0
	r7g.12xlarge	emr-5.36.1, emr-6.10.0, emr-7.0.0
	r7g.16xlarge	emr-5.36.1, emr-6.10.0, emr-7.0.0
	r7gd.xlarge	emr-5.36.1, emr-6.10.0, emr-7.0.0
	r7gd.2xlarge	emr-5.36.1, emr-6.10.0, emr-7.0.0
	r7gd.4xlarge	emr-5.36.1, emr-6.10.0, emr-7.0.0
	r7gd.8xlarge	emr-5.36.1, emr-6.10.0, emr-7.0.0
	r7gd.12xlarge	emr-5.36.1, emr-6.10.0, emr-7.0.0
	r7gd.16xlarge	emr-5.36.1, emr-6.10.0, emr-7.0.0
	x1.16xlarge	emr-5.36.1, emr-6.9.0, emr-7.0.0
	x1.32xlarge	emr-5.36.1, emr-6.9.0, emr-7.0.0
	x1e.xlarge	emr-5.36.1, emr-6.10.0, emr-7.0.0

Kelas Instans	Tipe instans	Versi EMR Amazon yang didukung minimum (5.x, 6.x, 7.x)
	x1e.2xlarge	emr-5.36.1, emr-6.10.0, emr-7.0.0
	x1e.4xlarge	emr-5.36.1, emr-6.10.0, emr-7.0.0
	x1e.8xlarge	emr-5.36.1, emr-6.10.0, emr-7.0.0
	x1e.16xlarge	emr-5.36.1, emr-6.10.0, emr-7.0.0
	x1e.32xlarge	emr-5.36.1, emr-6.10.0, emr-7.0.0
	x2idn.16xlarge	emr-5.36.1, emr-6.9.0
	x2idn.24xlarge	emr-5.36.1, emr-6.9.0
	x2idn.32xlarge	emr-5.36.1, emr-6.9.0
	x2iedn.xlarge	emr-5.36.1, emr-6.9.0
	x2iedn.2xlarge	emr-5.36.1, emr-6.9.0
	x2iedn.4xlarge	emr-5.36.1, emr-6.9.0
	x2iedn.8xlarge	emr-5.36.1, emr-6.9.0
	x2iedn.16xlarge	emr-5.36.1, emr-6.9.0
	x2iedn.24xlarge	emr-5.36.1, emr-6.9.0
	x2iedn.32xlarge	emr-5.36.1, emr-6.9.0
	z1d.xlarge	emr-5.17.0, emr-6.0.0, emr-7.0.0

Kelas Instans	Tipe instans	Versi EMR Amazon yang didukung minimum (5.x, 6.x, 7.x)
	z1d.2xlarge	emr-5.17.0, emr-6.0.0, emr-7.0.0
	z1d.3xlarge	emr-5.17.0, emr-6.0.0, emr-7.0.0
	z1d.6xlarge	emr-5.17.0, emr-6.0.0, emr-7.0.0
	z1d.12xlarge	emr-5.17.0, emr-6.0.0, emr-7.0.0
Penyimpanan Dioptimalkan	d3.xlarge	emr-5.33.0, emr-6.3.0, emr-7.0.0
	d3.2xlarge	emr-5.33.0, emr-6.3.0, emr-7.0.0
	d3.4xlarge	emr-5.33.0, emr-6.3.0, emr-7.0.0
	d3.8xlarge	emr-5.33.0, emr-6.3.0, emr-7.0.0
	i3.xlarge	emr-5.9.0, emr-6.0.0, emr-7.0.0
	i3.2xlarge	emr-5.9.0, emr-6.0.0, emr-7.0.0
	i3.4xlarge	emr-5.9.0, emr-6.0.0, emr-7.0.0
	i3.8xlarge	emr-5.9.0, emr-6.0.0, emr-7.0.0

Kelas Instans	Tipe instans	Versi EMR Amazon yang didukung minimum (5.x, 6.x, 7.x)
	i3.16xlarge	emr-5.9.0, emr-6.0.0, emr-7.0.0
	i3en.xlarge	emr-5.25.0, emr-6.0.0, emr-7.0.0
	i3en.2xlarge	emr-5.25.0, emr-6.0.0, emr-7.0.0
	i3en.3xlarge	emr-5.25.0, emr-6.0.0, emr-7.0.0
	i3en.6xlarge	emr-5.25.0, emr-6.0.0, emr-7.0.0
	i3en.12xlarge	emr-5.25.0, emr-6.0.0, emr-7.0.0
	i3en.24xlarge	emr-5.25.0, emr-6.0.0, emr-7.0.0
	i4i.xlarge	emr-5.36.1, emr-6.8.0
	i4i.2xlarge	emr-5.36.1, emr-6.8.0
	i4i.4xlarge	emr-5.36.1, emr-6.8.0
	i4i.8xlarge	emr-5.36.1, emr-6.8.0
	i4i.12xlarge	emr-5.36.1, emr-6.10.0
	i4i.16xlarge	emr-5.36.1, emr-6.8.0
	i4i.24xlarge	emr-5.36.1, emr-6.10.0
	i4i.32xlarge	emr-5.36.1, emr-6.8.0

Kelas Instans	Tipe instans	Versi EMR Amazon yang didukung minimum (5.x, 6.x, 7.x)
	is4gen.xlarge	emr-5.35.0, emr-6.6.0, emr-7.0.0
	is4gen.2xlarge	emr-5.35.0, emr-6.6.0, emr-7.0.0
	is4gen.4xlarge	emr-5.35.0, emr-6.6.0, emr-7.0.0
	is4gen.8xlarge	emr-5.35.0, emr-6.6.0, emr-7.0.0

Asia Pasifik (Hyderabad) - ap-selatan-2

Kelas Instans	Tipe instans	Versi EMR Amazon yang didukung minimum (5.x, 6.x, 7.x)
Tujuan Umum	m5.xlarge	emr-5.36.0, emr-6.7.0, emr-7.0.0
	m5.2xlarge	emr-5.36.0, emr-6.7.0, emr-7.0.0
	m5.4xlarge	emr-5.36.0, emr-6.7.0, emr-7.0.0
	m5.8xlarge	emr-5.36.0, emr-6.7.0, emr-7.0.0
	m5.12xlarge	emr-5.36.0, emr-6.7.0, emr-7.0.0

Kelas Instans	Tipe instans	Versi EMR Amazon yang didukung minimum (5.x, 6.x, 7.x)
	m5.16xlarge	emr-5.36.0, emr-6.7.0, emr-7.0.0
	m5.24xlarge	emr-5.36.0, emr-6.7.0, emr-7.0.0
	m5d.xlarge	emr-5.36.0, emr-6.7.0, emr-7.0.0
	m5d.2xlarge	emr-5.36.0, emr-6.7.0, emr-7.0.0
	m5d.4xlarge	emr-5.36.0, emr-6.7.0, emr-7.0.0
	m5d.8xlarge	emr-5.36.0, emr-6.7.0, emr-7.0.0
	m5d.12xlarge	emr-5.36.0, emr-6.7.0, emr-7.0.0
	m5d.16xlarge	emr-5.36.0, emr-6.7.0, emr-7.0.0
	m5d.24xlarge	emr-5.36.0, emr-6.7.0, emr-7.0.0
	m6a.xlarge	emr-5.36.1, emr-6.8.0, emr-7.0.0
	m6a.2xlarge	emr-5.36.1, emr-6.8.0, emr-7.0.0
	m6a.4xlarge	emr-5.36.1, emr-6.8.0, emr-7.0.0

Kelas Instans	Tipe instans	Versi EMR Amazon yang didukung minimum (5.x, 6.x, 7.x)
	m6a.8xlarge	emr-5.36.1, emr-6.8.0, emr-7.0.0
	m6a.12xlarge	emr-5.36.1, emr-6.8.0, emr-7.0.0
	m6a.16xlarge	emr-5.36.1, emr-6.8.0, emr-7.0.0
	m6a.24xlarge	emr-5.36.1, emr-6.8.0, emr-7.0.0
	m6a.32xlarge	emr-5.36.1, emr-6.8.0, emr-7.0.0
	m6a.48xlarge	emr-5.36.1, emr-6.8.0, emr-7.0.0
	m6g.xlarge	emr-5.36.0, emr-6.7.0, emr-7.0.0
	m6g.2xlarge	emr-5.36.0, emr-6.7.0, emr-7.0.0
	m6g.4xlarge	emr-5.36.0, emr-6.7.0, emr-7.0.0
	m6g.8xlarge	emr-5.36.0, emr-6.7.0, emr-7.0.0
	m6g.12xlarge	emr-5.36.0, emr-6.7.0, emr-7.0.0
	m6g.16xlarge	emr-5.36.0, emr-6.7.0, emr-7.0.0

Kelas Instans	Tipe instans	Versi EMR Amazon yang didukung minimum (5.x, 6.x, 7.x)
	m6gd.xlarge	emr-5.36.0, emr-6.7.0, emr-7.0.0
	m6gd.2xlarge	emr-5.36.0, emr-6.7.0, emr-7.0.0
	m6gd.4xlarge	emr-5.36.0, emr-6.7.0, emr-7.0.0
	m6gd.8xlarge	emr-5.36.0, emr-6.7.0, emr-7.0.0
	m6gd.12xlarge	emr-5.36.0, emr-6.7.0, emr-7.0.0
	m6gd.16xlarge	emr-5.36.0, emr-6.7.0, emr-7.0.0
	m6i.xlarge	emr-5.36.0, emr-6.7.0
	m6i.2xlarge	emr-5.36.0, emr-6.7.0
	m6i.4xlarge	emr-5.36.0, emr-6.7.0
	m6i.8xlarge	emr-5.36.0, emr-6.7.0
	m6i.12xlarge	emr-5.36.0, emr-6.7.0
	m6i.16xlarge	emr-5.36.0, emr-6.7.0
	m6i.24xlarge	emr-5.36.0, emr-6.7.0
	m6i.32xlarge	emr-5.36.0, emr-6.7.0
	m7g.xlarge	emr-5.36.1, emr-6.10.0, emr-7.0.0

Kelas Instans	Tipe instans	Versi EMR Amazon yang didukung minimum (5.x, 6.x, 7.x)
	m7g.2xlarge	emr-5.36.1, emr-6.10.0, emr-7.0.0
	m7g.4xlarge	emr-5.36.1, emr-6.10.0, emr-7.0.0
	m7g.8xlarge	emr-5.36.1, emr-6.10.0, emr-7.0.0
	m7g.12xlarge	emr-5.36.1, emr-6.10.0, emr-7.0.0
	m7g.16xlarge	emr-5.36.1, emr-6.10.0, emr-7.0.0
Komputasi Dioptimalkan	c5.xlarge	emr-5.36.0, emr-6.7.0, emr-7.0.0
	c5.2xlarge	emr-5.36.0, emr-6.7.0, emr-7.0.0
	c5.4xlarge	emr-5.36.0, emr-6.7.0, emr-7.0.0
	c5.9xlarge	emr-5.36.0, emr-6.7.0, emr-7.0.0
	c5.12xlarge	emr-5.36.0, emr-6.7.0, emr-7.0.0
	c5.18xlarge	emr-5.36.0, emr-6.7.0, emr-7.0.0
	c5.24xlarge	emr-5.36.0, emr-6.7.0, emr-7.0.0

Kelas Instans	Tipe instans	Versi EMR Amazon yang didukung minimum (5.x, 6.x, 7.x)
	c5d.xlarge	emr-5.36.0, emr-6.7.0, emr-7.0.0
	c5d.2xlarge	emr-5.36.0, emr-6.7.0, emr-7.0.0
	c5d.4xlarge	emr-5.36.0, emr-6.7.0, emr-7.0.0
	c5d.9xlarge	emr-5.36.0, emr-6.7.0, emr-7.0.0
	c5d.12xlarge	emr-5.36.0, emr-6.7.0, emr-7.0.0
	c5d.18xlarge	emr-5.36.0, emr-6.7.0, emr-7.0.0
	c5d.24xlarge	emr-5.36.0, emr-6.7.0, emr-7.0.0
	c6g.xlarge	emr-5.36.0, emr-6.7.0, emr-7.0.0
	c6g.2xlarge	emr-5.36.0, emr-6.7.0, emr-7.0.0
	c6g.4xlarge	emr-5.36.0, emr-6.7.0, emr-7.0.0
	c6g.8xlarge	emr-5.36.0, emr-6.7.0, emr-7.0.0
	c6g.12xlarge	emr-5.36.0, emr-6.7.0, emr-7.0.0

Kelas Instans	Tipe instans	Versi EMR Amazon yang didukung minimum (5.x, 6.x, 7.x)
	c6g.16xlarge	emr-5.36.0, emr-6.7.0, emr-7.0.0
	c6i.xlarge	emr-5.36.0, emr-6.7.0
	c6i.2xlarge	emr-5.36.0, emr-6.7.0
	c6i.4xlarge	emr-5.36.0, emr-6.7.0
	c6i.8xlarge	emr-5.36.0, emr-6.7.0
	c6i.12xlarge	emr-5.36.0, emr-6.7.0
	c6i.16xlarge	emr-5.36.0, emr-6.7.0
	c6i.24xlarge	emr-5.36.0, emr-6.7.0
	c6i.32xlarge	emr-5.36.0, emr-6.7.0
	c7g.xlarge	emr-5.36.1, emr-6.7.0, emr-7.0.0
	c7g.2xlarge	emr-5.36.1, emr-6.7.0, emr-7.0.0
	c7g.4xlarge	emr-5.36.1, emr-6.7.0, emr-7.0.0
	c7g.8xlarge	emr-5.36.1, emr-6.7.0, emr-7.0.0
	c7g.12xlarge	emr-5.36.1, emr-6.7.0, emr-7.0.0
	c7g.16xlarge	emr-5.36.1, emr-6.7.0, emr-7.0.0

Kelas Instans	Tipe instans	Versi EMR Amazon yang didukung minimum (5.x, 6.x, 7.x)
Memori Dioptimalkan	r5.xlarge	emr-5.36.0, emr-6.7.0, emr-7.0.0
	r5.2xlarge	emr-5.36.0, emr-6.7.0, emr-7.0.0
	r5.4xlarge	emr-5.36.0, emr-6.7.0, emr-7.0.0
	r5.8xlarge	emr-5.36.0, emr-6.7.0, emr-7.0.0
	r5.12xlarge	emr-5.36.0, emr-6.7.0, emr-7.0.0
	r5.16xlarge	emr-5.36.0, emr-6.7.0, emr-7.0.0
	r5.24xlarge	emr-5.36.0, emr-6.7.0, emr-7.0.0
	r5d.xlarge	emr-5.36.0, emr-6.7.0, emr-7.0.0
	r5d.2xlarge	emr-5.36.0, emr-6.7.0, emr-7.0.0
	r5d.4xlarge	emr-5.36.0, emr-6.7.0, emr-7.0.0
	r5d.8xlarge	emr-5.36.0, emr-6.7.0, emr-7.0.0
	r5d.12xlarge	emr-5.36.0, emr-6.7.0, emr-7.0.0

Kelas Instans	Tipe instans	Versi EMR Amazon yang didukung minimum (5.x, 6.x, 7.x)
	r5d.16xlarge	emr-5.36.0, emr-6.7.0, emr-7.0.0
	r5d.24xlarge	emr-5.36.0, emr-6.7.0, emr-7.0.0
	r6g.xlarge	emr-5.36.0, emr-6.7.0, emr-7.0.0
	r6g.2xlarge	emr-5.36.0, emr-6.7.0, emr-7.0.0
	r6g.4xlarge	emr-5.36.0, emr-6.7.0, emr-7.0.0
	r6g.8xlarge	emr-5.36.0, emr-6.7.0, emr-7.0.0
	r6g.12xlarge	emr-5.36.0, emr-6.7.0, emr-7.0.0
	r6g.16xlarge	emr-5.36.0, emr-6.7.0, emr-7.0.0
	r6i.xlarge	emr-5.36.0, emr-6.7.0
	r6i.2xlarge	emr-5.36.0, emr-6.7.0
	r6i.4xlarge	emr-5.36.0, emr-6.7.0
	r6i.8xlarge	emr-5.36.0, emr-6.7.0
	r6i.12xlarge	emr-5.36.0, emr-6.7.0
	r6i.16xlarge	emr-5.36.0, emr-6.7.0

Kelas Instans	Tipe instans	Versi EMR Amazon yang didukung minimum (5.x, 6.x, 7.x)
	r6i.24xlarge	emr-5.36.0, emr-6.7.0
	r6i.32xlarge	emr-5.36.0, emr-6.7.0
	r7g.xlarge	emr-5.36.1, emr-6.10.0, emr-7.0.0
	r7g.2xlarge	emr-5.36.1, emr-6.10.0, emr-7.0.0
	r7g.4xlarge	emr-5.36.1, emr-6.10.0, emr-7.0.0
	r7g.8xlarge	emr-5.36.1, emr-6.10.0, emr-7.0.0
	r7g.12xlarge	emr-5.36.1, emr-6.10.0, emr-7.0.0
	r7g.16xlarge	emr-5.36.1, emr-6.10.0, emr-7.0.0
	x2idn.16xlarge	emr-5.36.1, emr-6.9.0
	x2idn.24xlarge	emr-5.36.1, emr-6.9.0
	x2idn.32xlarge	emr-5.36.1, emr-6.9.0
	x2iedn.xlarge	emr-5.36.1, emr-6.9.0
	x2iedn.2xlarge	emr-5.36.1, emr-6.9.0
	x2iedn.4xlarge	emr-5.36.1, emr-6.9.0
	x2iedn.8xlarge	emr-5.36.1, emr-6.9.0
	x2iedn.16xlarge	emr-5.36.1, emr-6.9.0

Kelas Instans	Tipe instans	Versi EMR Amazon yang didukung minimum (5.x, 6.x, 7.x)
	x2iedn.24xlarge	emr-5.36.1, emr-6.9.0
	x2iedn.32xlarge	emr-5.36.1, emr-6.9.0
Penyimpanan Dioptimalkan	i3.xlarge	emr-5.36.0, emr-6.7.0, emr-7.0.0
	i3.2xlarge	emr-5.36.0, emr-6.7.0, emr-7.0.0
	i3.4xlarge	emr-5.36.0, emr-6.7.0, emr-7.0.0
	i3.8xlarge	emr-5.36.0, emr-6.7.0, emr-7.0.0
	i3.16xlarge	emr-5.36.0, emr-6.7.0, emr-7.0.0
	i3en.xlarge	emr-5.36.0, emr-6.7.0, emr-7.0.0
	i3en.2xlarge	emr-5.36.0, emr-6.7.0, emr-7.0.0
	i3en.3xlarge	emr-5.36.0, emr-6.7.0, emr-7.0.0
	i3en.6xlarge	emr-5.36.0, emr-6.7.0, emr-7.0.0
	i3en.12xlarge	emr-5.36.0, emr-6.7.0, emr-7.0.0
	i3en.24xlarge	emr-5.36.0, emr-6.7.0, emr-7.0.0

Kelas Instans	Tipe instans	Versi EMR Amazon yang didukung minimum (5.x, 6.x, 7.x)
	i4i.xlarge	emr-5.36.1, emr-6.8.0
	i4i.2xlarge	emr-5.36.1, emr-6.8.0
	i4i.4xlarge	emr-5.36.1, emr-6.8.0
	i4i.8xlarge	emr-5.36.1, emr-6.8.0
	i4i.12xlarge	emr-5.36.1, emr-6.10.0
	i4i.16xlarge	emr-5.36.1, emr-6.8.0
	i4i.24xlarge	emr-5.36.1, emr-6.10.0
	i4i.32xlarge	emr-5.36.1, emr-6.8.0

Asia Pasifik (Osaka) - ap-timur laut-3

Kelas Instans	Tipe instans	Versi EMR Amazon yang didukung minimum (5.x, 6.x, 7.x)
Tujuan Umum	m5.xlarge	emr-5.13.0, emr-6.0.0
	m5.2xlarge	emr-5.13.0, emr-6.0.0
	m5.4xlarge	emr-5.13.0, emr-6.0.0
	m5.8xlarge	emr-5.13.0, emr-6.0.0
	m5.12xlarge	emr-5.13.0, emr-6.0.0
	m5.16xlarge	emr-5.13.0, emr-6.0.0
	m5.24xlarge	emr-5.13.0, emr-6.0.0

Kelas Instans	Tipe instans	Versi EMR Amazon yang didukung minimum (5.x, 6.x, 7.x)
	m5d.xlarge	emr-5.13.0, emr-6.0.0
	m5d.2xlarge	emr-5.13.0, emr-6.0.0
	m5d.4xlarge	emr-5.13.0, emr-6.0.0
	m5d.8xlarge	emr-5.13.0, emr-6.0.0
	m5d.12xlarge	emr-5.13.0, emr-6.0.0
	m5d.16xlarge	emr-5.13.0, emr-6.0.0
	m5d.24xlarge	emr-5.13.0, emr-6.0.0
	m6g.xlarge	emr-5.30.0, emr-6.1.0
	m6g.2xlarge	emr-5.30.0, emr-6.1.0
	m6g.4xlarge	emr-5.30.0, emr-6.1.0
	m6g.8xlarge	emr-5.30.0, emr-6.1.0
	m6g.12xlarge	emr-5.30.0, emr-6.1.0
	m6g.16xlarge	emr-5.30.0, emr-6.1.0
	m6gd.xlarge	emr-5.33.0, emr-6.3.0
	m6gd.2xlarge	emr-5.33.0, emr-6.3.0
	m6gd.4xlarge	emr-5.33.0, emr-6.3.0
	m6gd.8xlarge	emr-5.33.0, emr-6.3.0
	m6gd.12xlarge	emr-5.33.0, emr-6.3.0
	m6gd.16xlarge	emr-5.33.0, emr-6.3.0

Kelas Instans	Tipe instans	Versi EMR Amazon yang didukung minimum (5.x, 6.x, 7.x)
	m6i.xlarge	emr-5.35.0, emr-6.6.0
	m6i.2xlarge	emr-5.35.0, emr-6.6.0
	m6i.4xlarge	emr-5.35.0, emr-6.6.0
	m6i.8xlarge	emr-5.35.0, emr-6.6.0
	m6i.12xlarge	emr-5.35.0, emr-6.6.0
	m6i.16xlarge	emr-5.35.0, emr-6.6.0
	m6i.24xlarge	emr-5.35.0, emr-6.6.0
	m6i.32xlarge	emr-5.35.0, emr-6.6.0
Komputasi Dioptimalkan	c5.xlarge	emr-5.13.0, emr-6.0.0
	c5.2xlarge	emr-5.13.0, emr-6.0.0
	c5.4xlarge	emr-5.13.0, emr-6.0.0
	c5.9xlarge	emr-5.13.0, emr-6.0.0
	c5.12xlarge	emr-5.13.0, emr-6.0.0
	c5.18xlarge	emr-5.13.0, emr-6.0.0
	c5.24xlarge	emr-5.13.0, emr-6.0.0
	c5d.xlarge	emr-5.13.0, emr-6.0.0
	c5d.2xlarge	emr-5.13.0, emr-6.0.0
	c5d.4xlarge	emr-5.13.0, emr-6.0.0
	c5d.9xlarge	emr-5.13.0, emr-6.0.0

Kelas Instans	Tipe instans	Versi EMR Amazon yang didukung minimum (5.x, 6.x, 7.x)
	c5d.12xlarge	emr-5.13.0, emr-6.0.0
	c5d.18xlarge	emr-5.13.0, emr-6.0.0
	c5d.24xlarge	emr-5.13.0, emr-6.0.0
	c5n.xlarge	emr-5.20.0, emr-6.0.0
	c5n.2xlarge	emr-5.20.0, emr-6.0.0
	c5n.4xlarge	emr-5.20.0, emr-6.0.0
	c5n.9xlarge	emr-5.20.0, emr-6.0.0
	c5n.18xlarge	emr-5.20.0, emr-6.0.0
	c6g.xlarge	emr-5.31.0, emr-6.1.0
	c6g.2xlarge	emr-5.31.0, emr-6.1.0
	c6g.4xlarge	emr-5.31.0, emr-6.1.0
	c6g.8xlarge	emr-5.31.0, emr-6.1.0
	c6g.12xlarge	emr-5.31.0, emr-6.1.0
	c6g.16xlarge	emr-5.31.0, emr-6.1.0
	c6gd.xlarge	emr-5.33.0, emr-6.3.0
	c6gd.2xlarge	emr-5.33.0, emr-6.3.0
	c6gd.4xlarge	emr-5.33.0, emr-6.3.0
	c6gd.8xlarge	emr-5.33.0, emr-6.3.0
	c6gd.12xlarge	emr-5.33.0, emr-6.3.0

Kelas Instans	Tipe instans	Versi EMR Amazon yang didukung minimum (5.x, 6.x, 7.x)
	c6gd.16xlarge	emr-5.33.0, emr-6.3.0
	c6gn.xlarge	emr-5.33.0, emr-6.3.0
	c6gn.2xlarge	emr-5.33.0, emr-6.3.0
	c6gn.4xlarge	emr-5.33.0, emr-6.3.0
	c6gn.8xlarge	emr-5.33.0, emr-6.3.0
	c6gn.12xlarge	emr-5.33.0, emr-6.3.0
	c6gn.16xlarge	emr-5.33.0, emr-6.3.0
	c6i.xlarge	emr-5.35.0, emr-6.6.0
	c6i.2xlarge	emr-5.35.0, emr-6.6.0
	c6i.4xlarge	emr-5.35.0, emr-6.6.0
	c6i.8xlarge	emr-5.35.0, emr-6.6.0
	c6i.12xlarge	emr-5.35.0, emr-6.6.0
	c6i.16xlarge	emr-5.35.0, emr-6.6.0
	c6i.24xlarge	emr-5.35.0, emr-6.6.0
	c6i.32xlarge	emr-5.35.0, emr-6.6.0
Komputasi yang Dipercepat	g4dn.xlarge	emr-5.30.0, emr-6.0.0
	g4dn.2xlarge	emr-5.30.0, emr-6.0.0
	g4dn.4xlarge	emr-5.30.0, emr-6.0.0
	g4dn.8xlarge	emr-5.30.0, emr-6.0.0

Kelas Instans	Tipe instans	Versi EMR Amazon yang didukung minimum (5.x, 6.x, 7.x)
	g4dn.12xlarge	emr-5.30.0, emr-6.0.0
	g4dn.16xlarge	emr-5.30.0, emr-6.0.0
Memori Dioptimalkan	r5.xlarge	emr-5.13.0, emr-6.0.0
	r5.2xlarge	emr-5.13.0, emr-6.0.0
	r5.4xlarge	emr-5.13.0, emr-6.0.0
	r5.8xlarge	emr-5.13.0, emr-6.0.0
	r5.12xlarge	emr-5.13.0, emr-6.0.0
	r5.16xlarge	emr-5.13.0, emr-6.0.0
	r5.24xlarge	emr-5.13.0, emr-6.0.0
	r5d.xlarge	emr-5.13.0, emr-6.0.0
	r5d.2xlarge	emr-5.13.0, emr-6.0.0
	r5d.4xlarge	emr-5.13.0, emr-6.0.0
	r5d.8xlarge	emr-5.13.0, emr-6.0.0
	r5d.12xlarge	emr-5.13.0, emr-6.0.0
	r5d.16xlarge	emr-5.13.0, emr-6.0.0
	r5d.24xlarge	emr-5.13.0, emr-6.0.0
	r6g.xlarge	emr-5.31.0, emr-6.1.0
	r6g.2xlarge	emr-5.31.0, emr-6.1.0
	r6g.4xlarge	emr-5.31.0, emr-6.1.0

Kelas Instans	Tipe instans	Versi EMR Amazon yang didukung minimum (5.x, 6.x, 7.x)
	r6g.8xlarge	emr-5.31.0, emr-6.1.0
	r6g.12xlarge	emr-5.31.0, emr-6.1.0
	r6g.16xlarge	emr-5.31.0, emr-6.1.0
	r6gd.xlarge	emr-5.33.0, emr-6.3.0
	r6gd.2xlarge	emr-5.33.0, emr-6.3.0
	r6gd.4xlarge	emr-5.33.0, emr-6.3.0
	r6gd.8xlarge	emr-5.33.0, emr-6.3.0
	r6gd.12xlarge	emr-5.33.0, emr-6.3.0
	r6gd.16xlarge	emr-5.33.0, emr-6.3.0
	r6i.xlarge	emr-5.35.0, emr-6.6.0
	r6i.2xlarge	emr-5.35.0, emr-6.6.0
	r6i.4xlarge	emr-5.35.0, emr-6.6.0
	r6i.8xlarge	emr-5.35.0, emr-6.6.0
	r6i.12xlarge	emr-5.35.0, emr-6.6.0
	r6i.16xlarge	emr-5.35.0, emr-6.6.0
	r6i.24xlarge	emr-5.35.0, emr-6.6.0
	r6i.32xlarge	emr-5.35.0, emr-6.6.0
	x1.16xlarge	emr-5.36.1, emr-6.9.0
	x1.32xlarge	emr-5.36.1, emr-6.9.0

Kelas Instans	Tipe instans	Versi EMR Amazon yang didukung minimum (5.x, 6.x, 7.x)
	x1e.xlarge	emr-5.36.1, emr-6.10.0
	x1e.2xlarge	emr-5.36.1, emr-6.10.0
	x1e.4xlarge	emr-5.36.1, emr-6.10.0
	x1e.8xlarge	emr-5.36.1, emr-6.10.0
	x1e.16xlarge	emr-5.36.1, emr-6.10.0
	x1e.32xlarge	emr-5.36.1, emr-6.10.0
	x2idn.16xlarge	emr-5.36.1, emr-6.9.0
	x2idn.24xlarge	emr-5.36.1, emr-6.9.0
	x2idn.32xlarge	emr-5.36.1, emr-6.9.0
	x2iedn.xlarge	emr-5.36.1, emr-6.9.0
	x2iedn.2xlarge	emr-5.36.1, emr-6.9.0
	x2iedn.4xlarge	emr-5.36.1, emr-6.9.0
	x2iedn.8xlarge	emr-5.36.1, emr-6.9.0
	x2iedn.16xlarge	emr-5.36.1, emr-6.9.0
	x2iedn.24xlarge	emr-5.36.1, emr-6.9.0
	x2iedn.32xlarge	emr-5.36.1, emr-6.9.0
Penyimpanan Dioptimalkan	i3.xlarge	emr-5.10.0, emr-6.0.0
	i3.2xlarge	emr-5.10.0, emr-6.0.0
	i3.4xlarge	emr-5.10.0, emr-6.0.0

Kelas Instans	Tipe instans	Versi EMR Amazon yang didukung minimum (5.x, 6.x, 7.x)
	i3.8xlarge	emr-5.10.0, emr-6.0.0
	i3.16xlarge	emr-5.10.0, emr-6.0.0
	i3en.xlarge	emr-5.25.0, emr-6.0.0
	i3en.2xlarge	emr-5.25.0, emr-6.0.0
	i3en.3xlarge	emr-5.25.0, emr-6.0.0
	i3en.6xlarge	emr-5.25.0, emr-6.0.0
	i3en.12xlarge	emr-5.25.0, emr-6.0.0
	i3en.24xlarge	emr-5.25.0, emr-6.0.0
	i4i.xlarge	emr-5.36.1, emr-6.8.0
	i4i.2xlarge	emr-5.36.1, emr-6.8.0
	i4i.4xlarge	emr-5.36.1, emr-6.8.0
	i4i.8xlarge	emr-5.36.1, emr-6.8.0
	i4i.12xlarge	emr-5.36.1, emr-6.10.0
	i4i.16xlarge	emr-5.36.1, emr-6.8.0
	i4i.24xlarge	emr-5.36.1, emr-6.10.0
	i4i.32xlarge	emr-5.36.1, emr-6.8.0

Asia Pacific (Seoul) - ap-northeast-2

Kelas Instans	Tipe instans	Versi EMR Amazon yang didukung minimum (5.x, 6.x, 7.x)
Tujuan Umum	m5.xlarge	emr-5.13.0, emr-6.0.0, emr-7.0.0
	m5.2xlarge	emr-5.13.0, emr-6.0.0, emr-7.0.0
	m5.4xlarge	emr-5.13.0, emr-6.0.0, emr-7.0.0
	m5.8xlarge	emr-5.13.0, emr-6.0.0, emr-7.0.0
	m5.12xlarge	emr-5.13.0, emr-6.0.0, emr-7.0.0
	m5.16xlarge	emr-5.13.0, emr-6.0.0, emr-7.0.0
	m5.24xlarge	emr-5.13.0, emr-6.0.0, emr-7.0.0
	m5a.xlarge	emr-5.20.0, emr-6.0.0, emr-7.0.0
	m5a.2xlarge	emr-5.20.0, emr-6.0.0, emr-7.0.0
	m5a.4xlarge	emr-5.20.0, emr-6.0.0, emr-7.0.0
	m5a.8xlarge	emr-5.20.0, emr-6.0.0, emr-7.0.0

Kelas Instans	Tipe instans	Versi EMR Amazon yang didukung minimum (5.x, 6.x, 7.x)
	m5a.12xlarge	emr-5.20.0, emr-6.0.0, emr-7.0.0
	m5a.16xlarge	emr-5.20.0, emr-6.0.0, emr-7.0.0
	m5a.24xlarge	emr-5.20.0, emr-6.0.0, emr-7.0.0
	m5ad.xlarge	emr-5.20.0, emr-6.0.0, emr-7.0.0
	m5ad.2xlarge	emr-5.20.0, emr-6.0.0, emr-7.0.0
	m5ad.4xlarge	emr-5.20.0, emr-6.0.0, emr-7.0.0
	m5ad.8xlarge	emr-5.33.0, emr-6.3.0, emr-7.0.0
	m5ad.12xlarge	emr-5.20.0, emr-6.0.0, emr-7.0.0
	m5ad.16xlarge	emr-5.33.0, emr-6.3.0, emr-7.0.0
	m5ad.24xlarge	emr-5.20.0, emr-6.0.0, emr-7.0.0
	m5d.xlarge	emr-5.13.0, emr-6.0.0, emr-7.0.0
	m5d.2xlarge	emr-5.13.0, emr-6.0.0, emr-7.0.0

Kelas Instans	Tipe instans	Versi EMR Amazon yang didukung minimum (5.x, 6.x, 7.x)
	m5d.4xlarge	emr-5.13.0, emr-6.0.0, emr-7.0.0
	m5d.8xlarge	emr-5.13.0, emr-6.0.0, emr-7.0.0
	m5d.12xlarge	emr-5.13.0, emr-6.0.0, emr-7.0.0
	m5d.16xlarge	emr-5.13.0, emr-6.0.0, emr-7.0.0
	m5d.24xlarge	emr-5.13.0, emr-6.0.0, emr-7.0.0
	m5zn.xlarge	emr-5.33.0, emr-6.3.0, emr-7.0.0
	m5zn.2xlarge	emr-5.33.0, emr-6.3.0, emr-7.0.0
	m5zn.3xlarge	emr-5.33.0, emr-6.3.0, emr-7.0.0
	m5zn.6xlarge	emr-5.33.0, emr-6.3.0, emr-7.0.0
	m5zn.12xlarge	emr-5.33.0, emr-6.3.0, emr-7.0.0
	m6g.xlarge	emr-5.30.0, emr-6.1.0, emr-7.0.0
	m6g.2xlarge	emr-5.30.0, emr-6.1.0, emr-7.0.0

Kelas Instans	Tipe instans	Versi EMR Amazon yang didukung minimum (5.x, 6.x, 7.x)
	m6g.4xlarge	emr-5.30.0, emr-6.1.0, emr-7.0.0
	m6g.8xlarge	emr-5.30.0, emr-6.1.0, emr-7.0.0
	m6g.12xlarge	emr-5.30.0, emr-6.1.0, emr-7.0.0
	m6g.16xlarge	emr-5.30.0, emr-6.1.0, emr-7.0.0
	m6gd.xlarge	emr-5.33.0, emr-6.3.0, emr-7.0.0
	m6gd.2xlarge	emr-5.33.0, emr-6.3.0, emr-7.0.0
	m6gd.4xlarge	emr-5.33.0, emr-6.3.0, emr-7.0.0
	m6gd.8xlarge	emr-5.33.0, emr-6.3.0, emr-7.0.0
	m6gd.12xlarge	emr-5.33.0, emr-6.3.0, emr-7.0.0
	m6gd.16xlarge	emr-5.33.0, emr-6.3.0, emr-7.0.0
	m6i.xlarge	emr-5.35.0, emr-6.6.0
	m6i.2xlarge	emr-5.35.0, emr-6.6.0
	m6i.4xlarge	emr-5.35.0, emr-6.6.0

Kelas Instans	Tipe instans	Versi EMR Amazon yang didukung minimum (5.x, 6.x, 7.x)
	m6i.8xlarge	emr-5.35.0, emr-6.6.0
	m6i.12xlarge	emr-5.35.0, emr-6.6.0
	m6i.16xlarge	emr-5.35.0, emr-6.6.0
	m6i.24xlarge	emr-5.35.0, emr-6.6.0
	m6i.32xlarge	emr-5.35.0, emr-6.6.0
	m7g.xlarge	emr-5.36.1, emr-6.10.0, emr-7.0.0
	m7g.2xlarge	emr-5.36.1, emr-6.10.0, emr-7.0.0
	m7g.4xlarge	emr-5.36.1, emr-6.10.0, emr-7.0.0
	m7g.8xlarge	emr-5.36.1, emr-6.10.0, emr-7.0.0
	m7g.12xlarge	emr-5.36.1, emr-6.10.0, emr-7.0.0
	m7g.16xlarge	emr-5.36.1, emr-6.10.0, emr-7.0.0
Komputasi Dioptimalkan	c5.xlarge	emr-5.13.0, emr-6.0.0, emr-7.0.0
	c5.2xlarge	emr-5.13.0, emr-6.0.0, emr-7.0.0
	c5.4xlarge	emr-5.13.0, emr-6.0.0, emr-7.0.0

Kelas Instans	Tipe instans	Versi EMR Amazon yang didukung minimum (5.x, 6.x, 7.x)
	c5.9xlarge	emr-5.13.0, emr-6.0.0, emr-7.0.0
	c5.12xlarge	emr-5.13.0, emr-6.0.0, emr-7.0.0
	c5.18xlarge	emr-5.13.0, emr-6.0.0, emr-7.0.0
	c5.24xlarge	emr-5.13.0, emr-6.0.0, emr-7.0.0
	c5a.xlarge	emr-5.31.0, emr-6.1.0, emr-7.0.0
	c5a.2xlarge	emr-5.31.0, emr-6.1.0, emr-7.0.0
	c5a.4xlarge	emr-5.31.0, emr-6.1.0, emr-7.0.0
	c5a.8xlarge	emr-5.31.0, emr-6.1.0, emr-7.0.0
	c5a.12xlarge	emr-5.31.0, emr-6.1.0, emr-7.0.0
	c5a.16xlarge	emr-5.31.0, emr-6.1.0, emr-7.0.0
	c5a.24xlarge	emr-5.31.0, emr-6.1.0, emr-7.0.0
	c5d.xlarge	emr-5.13.0, emr-6.0.0, emr-7.0.0

Kelas Instans	Tipe instans	Versi EMR Amazon yang didukung minimum (5.x, 6.x, 7.x)
	c5d.2xlarge	emr-5.13.0, emr-6.0.0, emr-7.0.0
	c5d.4xlarge	emr-5.13.0, emr-6.0.0, emr-7.0.0
	c5d.9xlarge	emr-5.13.0, emr-6.0.0, emr-7.0.0
	c5d.12xlarge	emr-5.13.0, emr-6.0.0, emr-7.0.0
	c5d.18xlarge	emr-5.13.0, emr-6.0.0, emr-7.0.0
	c5d.24xlarge	emr-5.13.0, emr-6.0.0, emr-7.0.0
	c5n.xlarge	emr-5.20.0, emr-6.0.0, emr-7.0.0
	c5n.2xlarge	emr-5.20.0, emr-6.0.0, emr-7.0.0
	c5n.4xlarge	emr-5.20.0, emr-6.0.0, emr-7.0.0
	c5n.9xlarge	emr-5.20.0, emr-6.0.0, emr-7.0.0
	c5n.18xlarge	emr-5.20.0, emr-6.0.0, emr-7.0.0
	c6g.xlarge	emr-5.31.0, emr-6.1.0, emr-7.0.0

Kelas Instans	Tipe instans	Versi EMR Amazon yang didukung minimum (5.x, 6.x, 7.x)
	c6g.2xlarge	emr-5.31.0, emr-6.1.0, emr-7.0.0
	c6g.4xlarge	emr-5.31.0, emr-6.1.0, emr-7.0.0
	c6g.8xlarge	emr-5.31.0, emr-6.1.0, emr-7.0.0
	c6g.12xlarge	emr-5.31.0, emr-6.1.0, emr-7.0.0
	c6g.16xlarge	emr-5.31.0, emr-6.1.0, emr-7.0.0
	c6gd.xlarge	emr-5.33.0, emr-6.3.0, emr-7.0.0
	c6gd.2xlarge	emr-5.33.0, emr-6.3.0, emr-7.0.0
	c6gd.4xlarge	emr-5.33.0, emr-6.3.0, emr-7.0.0
	c6gd.8xlarge	emr-5.33.0, emr-6.3.0, emr-7.0.0
	c6gd.12xlarge	emr-5.33.0, emr-6.3.0, emr-7.0.0
	c6gd.16xlarge	emr-5.33.0, emr-6.3.0, emr-7.0.0
	c6gn.xlarge	emr-5.33.0, emr-6.3.0, emr-7.0.0

Kelas Instans	Tipe instans	Versi EMR Amazon yang didukung minimum (5.x, 6.x, 7.x)
	c6gn.2xlarge	emr-5.33.0, emr-6.3.0, emr-7.0.0
	c6gn.4xlarge	emr-5.33.0, emr-6.3.0, emr-7.0.0
	c6gn.8xlarge	emr-5.33.0, emr-6.3.0, emr-7.0.0
	c6gn.12xlarge	emr-5.33.0, emr-6.3.0, emr-7.0.0
	c6gn.16xlarge	emr-5.33.0, emr-6.3.0, emr-7.0.0
	c6i.xlarge	emr-5.35.0, emr-6.6.0
	c6i.2xlarge	emr-5.35.0, emr-6.6.0
	c6i.4xlarge	emr-5.35.0, emr-6.6.0
	c6i.8xlarge	emr-5.35.0, emr-6.6.0
	c6i.12xlarge	emr-5.35.0, emr-6.6.0
	c6i.16xlarge	emr-5.35.0, emr-6.6.0
	c6i.24xlarge	emr-5.35.0, emr-6.6.0
	c6i.32xlarge	emr-5.35.0, emr-6.6.0
	c6in.xlarge	emr-5.36.1, emr-6.10.0
	c6in.2xlarge	emr-5.36.1, emr-6.10.0
	c6in.4xlarge	emr-5.36.1, emr-6.10.0

Kelas Instans	Tipe instans	Versi EMR Amazon yang didukung minimum (5.x, 6.x, 7.x)
	c6in.8xlarge	emr-5.36.1, emr-6.10.0
	c6in.12xlarge	emr-5.36.1, emr-6.10.0
	c6in.16xlarge	emr-5.36.1, emr-6.10.0
	c6in.24xlarge	emr-5.36.1, emr-6.10.0
	c6in.32xlarge	emr-5.36.1, emr-6.10.0
	c7g.xlarge	emr-5.36.1, emr-6.7.0, emr-7.0.0
	c7g.2xlarge	emr-5.36.1, emr-6.7.0, emr-7.0.0
	c7g.4xlarge	emr-5.36.1, emr-6.7.0, emr-7.0.0
	c7g.8xlarge	emr-5.36.1, emr-6.7.0, emr-7.0.0
	c7g.12xlarge	emr-5.36.1, emr-6.7.0, emr-7.0.0
	c7g.16xlarge	emr-5.36.1, emr-6.7.0, emr-7.0.0
Komputasi yang Dipercepat	g3.4xlarge	emr-5.18.0, emr-6.0.0, emr-7.0.0
	g3.8xlarge	emr-5.18.0, emr-6.0.0, emr-7.0.0
	g3.16xlarge	emr-5.18.0, emr-6.0.0, emr-7.0.0

Kelas Instans	Tipe instans	Versi EMR Amazon yang didukung minimum (5.x, 6.x, 7.x)
	g3s.xlarge	emr-5.19.0, emr-6.0.0, emr-7.0.0
	g4dn.xlarge	emr-5.30.0, emr-6.0.0, emr-7.0.0
	g4dn.2xlarge	emr-5.30.0, emr-6.0.0, emr-7.0.0
	g4dn.4xlarge	emr-5.30.0, emr-6.0.0, emr-7.0.0
	g4dn.8xlarge	emr-5.30.0, emr-6.0.0, emr-7.0.0
	g4dn.12xlarge	emr-5.30.0, emr-6.0.0, emr-7.0.0
	g4dn.16xlarge	emr-5.30.0, emr-6.0.0, emr-7.0.0
	g5.xlarge	emr-5.36.1, emr-6.9.0, emr-7.0.0
	g5.2xlarge	emr-5.36.1, emr-6.9.0, emr-7.0.0
	g5.4xlarge	emr-5.36.1, emr-6.9.0, emr-7.0.0
	g5.8xlarge	emr-5.36.1, emr-6.9.0, emr-7.0.0
	g5.12xlarge	emr-5.36.1, emr-6.9.0, emr-7.0.0

Kelas Instans	Tipe instans	Versi EMR Amazon yang didukung minimum (5.x, 6.x, 7.x)
	g5.16xlarge	emr-5.36.1, emr-6.9.0, emr-7.0.0
	g5.24xlarge	emr-5.36.1, emr-6.9.0, emr-7.0.0
	g5.48xlarge	emr-5.36.1, emr-6.9.0, emr-7.0.0
	p2.xlarge	emr-5.10.0, emr-6.0.0, emr-7.0.0
	p2.8xlarge	emr-5.10.0, emr-6.0.0, emr-7.0.0
	p2.16xlarge	emr-5.10.0, emr-6.0.0, emr-7.0.0
	p3.2xlarge	emr-5.10.0, emr-6.0.0, emr-7.0.0
	p3.8xlarge	emr-5.10.0, emr-6.0.0, emr-7.0.0
	p3.16xlarge	emr-5.10.0, emr-6.0.0, emr-7.0.0
Memori Dioptimalkan	r5.xlarge	emr-5.13.0, emr-6.0.0, emr-7.0.0
	r5.2xlarge	emr-5.13.0, emr-6.0.0, emr-7.0.0
	r5.4xlarge	emr-5.13.0, emr-6.0.0, emr-7.0.0

Kelas Instans	Tipe instans	Versi EMR Amazon yang didukung minimum (5.x, 6.x, 7.x)
	r5.8xlarge	emr-5.13.0, emr-6.0.0, emr-7.0.0
	r5.12xlarge	emr-5.13.0, emr-6.0.0, emr-7.0.0
	r5.16xlarge	emr-5.13.0, emr-6.0.0, emr-7.0.0
	r5.24xlarge	emr-5.13.0, emr-6.0.0, emr-7.0.0
	r5a.xlarge	emr-5.20.0, emr-6.0.0, emr-7.0.0
	r5a.2xlarge	emr-5.20.0, emr-6.0.0, emr-7.0.0
	r5a.4xlarge	emr-5.20.0, emr-6.0.0, emr-7.0.0
	r5a.8xlarge	emr-5.20.0, emr-6.0.0, emr-7.0.0
	r5a.12xlarge	emr-5.20.0, emr-6.0.0, emr-7.0.0
	r5a.16xlarge	emr-5.20.0, emr-6.0.0, emr-7.0.0
	r5a.24xlarge	emr-5.20.0, emr-6.0.0, emr-7.0.0
	r5ad.xlarge	emr-5.20.0, emr-6.0.0, emr-7.0.0

Kelas Instans	Tipe instans	Versi EMR Amazon yang didukung minimum (5.x, 6.x, 7.x)
	r5ad.2xlarge	emr-5.20.0, emr-6.0.0, emr-7.0.0
	r5ad.4xlarge	emr-5.20.0, emr-6.0.0, emr-7.0.0
	r5ad.8xlarge	emr-5.33.0, emr-6.3.0, emr-7.0.0
	r5ad.12xlarge	emr-5.20.0, emr-6.0.0, emr-7.0.0
	r5ad.16xlarge	emr-5.33.0, emr-6.3.0, emr-7.0.0
	r5ad.24xlarge	emr-5.20.0, emr-6.0.0, emr-7.0.0
	r5b.xlarge	emr-5.33.0, emr-6.3.0, emr-7.0.0
	r5b.2xlarge	emr-5.33.0, emr-6.3.0, emr-7.0.0
	r5b.4xlarge	emr-5.33.0, emr-6.3.0, emr-7.0.0
	r5b.8xlarge	emr-5.33.0, emr-6.3.0, emr-7.0.0
	r5b.12xlarge	emr-5.33.0, emr-6.3.0, emr-7.0.0
	r5b.16xlarge	emr-5.33.0, emr-6.3.0, emr-7.0.0

Kelas Instans	Tipe instans	Versi EMR Amazon yang didukung minimum (5.x, 6.x, 7.x)
	r5b.24xlarge	emr-5.33.0, emr-6.3.0, emr-7.0.0
	r5d.xlarge	emr-5.13.0, emr-6.0.0, emr-7.0.0
	r5d.2xlarge	emr-5.13.0, emr-6.0.0, emr-7.0.0
	r5d.4xlarge	emr-5.13.0, emr-6.0.0, emr-7.0.0
	r5d.8xlarge	emr-5.13.0, emr-6.0.0, emr-7.0.0
	r5d.12xlarge	emr-5.13.0, emr-6.0.0, emr-7.0.0
	r5d.16xlarge	emr-5.13.0, emr-6.0.0, emr-7.0.0
	r5d.24xlarge	emr-5.13.0, emr-6.0.0, emr-7.0.0
	r5dn.xlarge	emr-5.31.0, emr-6.1.0, emr-7.0.0
	r5dn.2xlarge	emr-5.31.0, emr-6.1.0, emr-7.0.0
	r5dn.4xlarge	emr-5.31.0, emr-6.1.0, emr-7.0.0
	r5dn.8xlarge	emr-5.31.0, emr-6.1.0, emr-7.0.0

Kelas Instans	Tipe instans	Versi EMR Amazon yang didukung minimum (5.x, 6.x, 7.x)
	r5dn.12xlarge	emr-5.31.0, emr-6.1.0, emr-7.0.0
	r5dn.16xlarge	emr-5.31.0, emr-6.1.0, emr-7.0.0
	r5dn.24xlarge	emr-5.31.0, emr-6.1.0, emr-7.0.0
	r5n.xlarge	emr-5.31.0, emr-6.1.0, emr-7.0.0
	r5n.2xlarge	emr-5.31.0, emr-6.1.0, emr-7.0.0
	r5n.4xlarge	emr-5.31.0, emr-6.1.0, emr-7.0.0
	r5n.8xlarge	emr-5.31.0, emr-6.1.0, emr-7.0.0
	r5n.12xlarge	emr-5.31.0, emr-6.1.0, emr-7.0.0
	r5n.16xlarge	emr-5.31.0, emr-6.1.0, emr-7.0.0
	r5n.24xlarge	emr-5.31.0, emr-6.1.0, emr-7.0.0
	r6g.xlarge	emr-5.31.0, emr-6.1.0, emr-7.0.0
	r6g.2xlarge	emr-5.31.0, emr-6.1.0, emr-7.0.0

Kelas Instans	Tipe instans	Versi EMR Amazon yang didukung minimum (5.x, 6.x, 7.x)
	r6g.4xlarge	emr-5.31.0, emr-6.1.0, emr-7.0.0
	r6g.8xlarge	emr-5.31.0, emr-6.1.0, emr-7.0.0
	r6g.12xlarge	emr-5.31.0, emr-6.1.0, emr-7.0.0
	r6g.16xlarge	emr-5.31.0, emr-6.1.0, emr-7.0.0
	r6gd.xlarge	emr-5.33.0, emr-6.3.0, emr-7.0.0
	r6gd.2xlarge	emr-5.33.0, emr-6.3.0, emr-7.0.0
	r6gd.4xlarge	emr-5.33.0, emr-6.3.0, emr-7.0.0
	r6gd.8xlarge	emr-5.33.0, emr-6.3.0, emr-7.0.0
	r6gd.12xlarge	emr-5.33.0, emr-6.3.0, emr-7.0.0
	r6gd.16xlarge	emr-5.33.0, emr-6.3.0, emr-7.0.0
	r6i.xlarge	emr-5.35.0, emr-6.6.0
	r6i.2xlarge	emr-5.35.0, emr-6.6.0
	r6i.4xlarge	emr-5.35.0, emr-6.6.0

Kelas Instans	Tipe instans	Versi EMR Amazon yang didukung minimum (5.x, 6.x, 7.x)
	r6i.8xlarge	emr-5.35.0, emr-6.6.0
	r6i.12xlarge	emr-5.35.0, emr-6.6.0
	r6i.16xlarge	emr-5.35.0, emr-6.6.0
	r6i.24xlarge	emr-5.35.0, emr-6.6.0
	r6i.32xlarge	emr-5.35.0, emr-6.6.0
	r6id.xlarge	emr-5.36.1, emr-6.8.0
	r6id.2xlarge	emr-5.36.1, emr-6.8.0
	r6id.4xlarge	emr-5.36.1, emr-6.8.0
	r6id.8xlarge	emr-5.36.1, emr-6.8.0
	r6id.12xlarge	emr-5.36.1, emr-6.8.0
	r6id.16xlarge	emr-5.36.1, emr-6.8.0
	r6id.24xlarge	emr-5.36.1, emr-6.8.0
	r6id.32xlarge	emr-5.36.1, emr-6.8.0
	r7g.xlarge	emr-5.36.1, emr-6.10.0, emr-7.0.0
	r7g.2xlarge	emr-5.36.1, emr-6.10.0, emr-7.0.0
	r7g.4xlarge	emr-5.36.1, emr-6.10.0, emr-7.0.0
	r7g.8xlarge	emr-5.36.1, emr-6.10.0, emr-7.0.0

Kelas Instans	Tipe instans	Versi EMR Amazon yang didukung minimum (5.x, 6.x, 7.x)
	r7g.12xlarge	emr-5.36.1, emr-6.10.0, emr-7.0.0
	r7g.16xlarge	emr-5.36.1, emr-6.10.0, emr-7.0.0
	x1.16xlarge	emr-5.36.1, emr-6.9.0, emr-7.0.0
	x1.32xlarge	emr-5.36.1, emr-6.9.0, emr-7.0.0
	x1e.xlarge	emr-5.36.1, emr-6.10.0, emr-7.0.0
	x1e.2xlarge	emr-5.36.1, emr-6.10.0, emr-7.0.0
	x1e.4xlarge	emr-5.36.1, emr-6.10.0, emr-7.0.0
	x1e.8xlarge	emr-5.36.1, emr-6.10.0, emr-7.0.0
	x1e.16xlarge	emr-5.36.1, emr-6.10.0, emr-7.0.0
	x1e.32xlarge	emr-5.36.1, emr-6.10.0, emr-7.0.0
	x2idn.16xlarge	emr-5.36.1, emr-6.9.0
	x2idn.24xlarge	emr-5.36.1, emr-6.9.0
	x2idn.32xlarge	emr-5.36.1, emr-6.9.0

Kelas Instans	Tipe instans	Versi EMR Amazon yang didukung minimum (5.x, 6.x, 7.x)
	x2iedn.xlarge	emr-5.36.1, emr-6.9.0
	x2iedn.2xlarge	emr-5.36.1, emr-6.9.0
	x2iedn.4xlarge	emr-5.36.1, emr-6.9.0
	x2iedn.8xlarge	emr-5.36.1, emr-6.9.0
	x2iedn.16xlarge	emr-5.36.1, emr-6.9.0
	x2iedn.24xlarge	emr-5.36.1, emr-6.9.0
	x2iedn.32xlarge	emr-5.36.1, emr-6.9.0
	z1d.xlarge	emr-5.17.0, emr-6.0.0, emr-7.0.0
	z1d.2xlarge	emr-5.17.0, emr-6.0.0, emr-7.0.0
	z1d.3xlarge	emr-5.17.0, emr-6.0.0, emr-7.0.0
	z1d.6xlarge	emr-5.17.0, emr-6.0.0, emr-7.0.0
	z1d.12xlarge	emr-5.17.0, emr-6.0.0, emr-7.0.0
Penyimpanan Dioptimalkan	i3.xlarge	emr-5.9.0, emr-6.0.0, emr-7.0.0
	i3.2xlarge	emr-5.9.0, emr-6.0.0, emr-7.0.0

Kelas Instans	Tipe instans	Versi EMR Amazon yang didukung minimum (5.x, 6.x, 7.x)
	i3.4xlarge	emr-5.9.0, emr-6.0.0, emr-7.0.0
	i3.8xlarge	emr-5.9.0, emr-6.0.0, emr-7.0.0
	i3.16xlarge	emr-5.9.0, emr-6.0.0, emr-7.0.0
	i3en.xlarge	emr-5.25.0, emr-6.0.0, emr-7.0.0
	i3en.2xlarge	emr-5.25.0, emr-6.0.0, emr-7.0.0
	i3en.3xlarge	emr-5.25.0, emr-6.0.0, emr-7.0.0
	i3en.6xlarge	emr-5.25.0, emr-6.0.0, emr-7.0.0
	i3en.12xlarge	emr-5.25.0, emr-6.0.0, emr-7.0.0
	i3en.24xlarge	emr-5.25.0, emr-6.0.0, emr-7.0.0
	i4i.xlarge	emr-5.36.1, emr-6.8.0
	i4i.2xlarge	emr-5.36.1, emr-6.8.0
	i4i.4xlarge	emr-5.36.1, emr-6.8.0
	i4i.8xlarge	emr-5.36.1, emr-6.8.0
	i4i.12xlarge	emr-5.36.1, emr-6.10.0

Kelas Instans	Tipe instans	Versi EMR Amazon yang didukung minimum (5.x, 6.x, 7.x)
	i4i.16xlarge	emr-5.36.1, emr-6.8.0
	i4i.24xlarge	emr-5.36.1, emr-6.10.0
	i4i.32xlarge	emr-5.36.1, emr-6.8.0

Asia Pacific (Singapore) - ap-southeast-1

Kelas Instans	Tipe instans	Versi EMR Amazon yang didukung minimum (5.x, 6.x, 7.x)
Tujuan Umum	m5.xlarge	emr-5.13.0, emr-6.0.0, emr-7.0.0
	m5.2xlarge	emr-5.13.0, emr-6.0.0, emr-7.0.0
	m5.4xlarge	emr-5.13.0, emr-6.0.0, emr-7.0.0
	m5.8xlarge	emr-5.13.0, emr-6.0.0, emr-7.0.0
	m5.12xlarge	emr-5.13.0, emr-6.0.0, emr-7.0.0
	m5.16xlarge	emr-5.13.0, emr-6.0.0, emr-7.0.0
	m5.24xlarge	emr-5.13.0, emr-6.0.0, emr-7.0.0

Kelas Instans	Tipe instans	Versi EMR Amazon yang didukung minimum (5.x, 6.x, 7.x)
	m5a.xlarge	emr-5.20.0, emr-6.0.0, emr-7.0.0
	m5a.2xlarge	emr-5.20.0, emr-6.0.0, emr-7.0.0
	m5a.4xlarge	emr-5.20.0, emr-6.0.0, emr-7.0.0
	m5a.8xlarge	emr-5.20.0, emr-6.0.0, emr-7.0.0
	m5a.12xlarge	emr-5.20.0, emr-6.0.0, emr-7.0.0
	m5a.16xlarge	emr-5.20.0, emr-6.0.0, emr-7.0.0
	m5a.24xlarge	emr-5.20.0, emr-6.0.0, emr-7.0.0
	m5ad.xlarge	emr-5.20.0, emr-6.0.0, emr-7.0.0
	m5ad.2xlarge	emr-5.20.0, emr-6.0.0, emr-7.0.0
	m5ad.4xlarge	emr-5.20.0, emr-6.0.0, emr-7.0.0
	m5ad.8xlarge	emr-5.33.0, emr-6.3.0, emr-7.0.0
	m5ad.12xlarge	emr-5.20.0, emr-6.0.0, emr-7.0.0

Kelas Instans	Tipe instans	Versi EMR Amazon yang didukung minimum (5.x, 6.x, 7.x)
	m5ad.16xlarge	emr-5.33.0, emr-6.3.0, emr-7.0.0
	m5ad.24xlarge	emr-5.20.0, emr-6.0.0, emr-7.0.0
	m5d.xlarge	emr-5.13.0, emr-6.0.0, emr-7.0.0
	m5d.2xlarge	emr-5.13.0, emr-6.0.0, emr-7.0.0
	m5d.4xlarge	emr-5.13.0, emr-6.0.0, emr-7.0.0
	m5d.8xlarge	emr-5.13.0, emr-6.0.0, emr-7.0.0
	m5d.12xlarge	emr-5.13.0, emr-6.0.0, emr-7.0.0
	m5d.16xlarge	emr-5.13.0, emr-6.0.0, emr-7.0.0
	m5d.24xlarge	emr-5.13.0, emr-6.0.0, emr-7.0.0
	m5dn.xlarge	emr-5.31.0, emr-6.1.0, emr-7.0.0
	m5dn.2xlarge	emr-5.31.0, emr-6.1.0, emr-7.0.0
	m5dn.4xlarge	emr-5.31.0, emr-6.1.0, emr-7.0.0

Kelas Instans	Tipe instans	Versi EMR Amazon yang didukung minimum (5.x, 6.x, 7.x)
	m5dn.8xlarge	emr-5.31.0, emr-6.1.0, emr-7.0.0
	m5dn.12xlarge	emr-5.31.0, emr-6.1.0, emr-7.0.0
	m5dn.16xlarge	emr-5.31.0, emr-6.1.0, emr-7.0.0
	m5dn.24xlarge	emr-5.31.0, emr-6.1.0, emr-7.0.0
	m5n.xlarge	emr-5.31.0, emr-6.1.0, emr-7.0.0
	m5n.2xlarge	emr-5.31.0, emr-6.1.0, emr-7.0.0
	m5n.4xlarge	emr-5.31.0, emr-6.1.0, emr-7.0.0
	m5n.8xlarge	emr-5.31.0, emr-6.1.0, emr-7.0.0
	m5n.12xlarge	emr-5.31.0, emr-6.1.0, emr-7.0.0
	m5n.16xlarge	emr-5.31.0, emr-6.1.0, emr-7.0.0
	m5n.24xlarge	emr-5.31.0, emr-6.1.0, emr-7.0.0
	m5zn.xlarge	emr-5.33.0, emr-6.3.0, emr-7.0.0

Kelas Instans	Tipe instans	Versi EMR Amazon yang didukung minimum (5.x, 6.x, 7.x)
	m5zn.2xlarge	emr-5.33.0, emr-6.3.0, emr-7.0.0
	m5zn.3xlarge	emr-5.33.0, emr-6.3.0, emr-7.0.0
	m5zn.6xlarge	emr-5.33.0, emr-6.3.0, emr-7.0.0
	m5zn.12xlarge	emr-5.33.0, emr-6.3.0, emr-7.0.0
	m6a.xlarge	emr-5.36.1, emr-6.8.0, emr-7.0.0
	m6a.2xlarge	emr-5.36.1, emr-6.8.0, emr-7.0.0
	m6a.4xlarge	emr-5.36.1, emr-6.8.0, emr-7.0.0
	m6a.8xlarge	emr-5.36.1, emr-6.8.0, emr-7.0.0
	m6a.12xlarge	emr-5.36.1, emr-6.8.0, emr-7.0.0
	m6a.16xlarge	emr-5.36.1, emr-6.8.0, emr-7.0.0
	m6a.24xlarge	emr-5.36.1, emr-6.8.0, emr-7.0.0
	m6a.32xlarge	emr-5.36.1, emr-6.8.0, emr-7.0.0

Kelas Instans	Tipe instans	Versi EMR Amazon yang didukung minimum (5.x, 6.x, 7.x)
	m6a.48xlarge	emr-5.36.1, emr-6.8.0, emr-7.0.0
	m6g.xlarge	emr-5.30.0, emr-6.1.0, emr-7.0.0
	m6g.2xlarge	emr-5.30.0, emr-6.1.0, emr-7.0.0
	m6g.4xlarge	emr-5.30.0, emr-6.1.0, emr-7.0.0
	m6g.8xlarge	emr-5.30.0, emr-6.1.0, emr-7.0.0
	m6g.12xlarge	emr-5.30.0, emr-6.1.0, emr-7.0.0
	m6g.16xlarge	emr-5.30.0, emr-6.1.0, emr-7.0.0
	m6gd.xlarge	emr-5.33.0, emr-6.3.0, emr-7.0.0
	m6gd.2xlarge	emr-5.33.0, emr-6.3.0, emr-7.0.0
	m6gd.4xlarge	emr-5.33.0, emr-6.3.0, emr-7.0.0
	m6gd.8xlarge	emr-5.33.0, emr-6.3.0, emr-7.0.0
	m6gd.12xlarge	emr-5.33.0, emr-6.3.0, emr-7.0.0

Kelas Instans	Tipe instans	Versi EMR Amazon yang didukung minimum (5.x, 6.x, 7.x)
	m6gd.16xlarge	emr-5.33.0, emr-6.3.0, emr-7.0.0
	m6i.xlarge	emr-5.35.0, emr-6.6.0
	m6i.2xlarge	emr-5.35.0, emr-6.6.0
	m6i.4xlarge	emr-5.35.0, emr-6.6.0
	m6i.8xlarge	emr-5.35.0, emr-6.6.0
	m6i.12xlarge	emr-5.35.0, emr-6.6.0
	m6i.16xlarge	emr-5.35.0, emr-6.6.0
	m6i.24xlarge	emr-5.35.0, emr-6.6.0
	m6i.32xlarge	emr-5.35.0, emr-6.6.0
	m6idn.xlarge	emr-5.36.1, emr-6.10.0
	m6idn.2xlarge	emr-5.36.1, emr-6.10.0
	m6idn.4xlarge	emr-5.36.1, emr-6.10.0
	m6idn.8xlarge	emr-5.36.1, emr-6.10.0
	m6idn.12xlarge	emr-5.36.1, emr-6.10.0
	m6idn.16xlarge	emr-5.36.1, emr-6.10.0
	m6idn.24xlarge	emr-5.36.1, emr-6.10.0
	m6idn.32xlarge	emr-5.36.1, emr-6.10.0
	m6in.xlarge	emr-5.36.1, emr-6.10.0
	m6in.2xlarge	emr-5.36.1, emr-6.10.0

Kelas Instans	Tipe instans	Versi EMR Amazon yang didukung minimum (5.x, 6.x, 7.x)
	m6in.4xlarge	emr-5.36.1, emr-6.10.0
	m6in.8xlarge	emr-5.36.1, emr-6.10.0
	m6in.12xlarge	emr-5.36.1, emr-6.10.0
	m6in.16xlarge	emr-5.36.1, emr-6.10.0
	m6in.24xlarge	emr-5.36.1, emr-6.10.0
	m6in.32xlarge	emr-5.36.1, emr-6.10.0
	m7g.xlarge	emr-5.36.1, emr-6.10.0, emr-7.0.0
	m7g.2xlarge	emr-5.36.1, emr-6.10.0, emr-7.0.0
	m7g.4xlarge	emr-5.36.1, emr-6.10.0, emr-7.0.0
	m7g.8xlarge	emr-5.36.1, emr-6.10.0, emr-7.0.0
	m7g.12xlarge	emr-5.36.1, emr-6.10.0, emr-7.0.0
	m7g.16xlarge	emr-5.36.1, emr-6.10.0, emr-7.0.0
	m7gd.xlarge	emr-5.36.1, emr-6.10.0, emr-7.0.0
	m7gd.2xlarge	emr-5.36.1, emr-6.10.0, emr-7.0.0

Kelas Instans	Tipe instans	Versi EMR Amazon yang didukung minimum (5.x, 6.x, 7.x)
	m7gd.4xlarge	emr-5.36.1, emr-6.10.0, emr-7.0.0
	m7gd.8xlarge	emr-5.36.1, emr-6.10.0, emr-7.0.0
	m7gd.12xlarge	emr-5.36.1, emr-6.10.0, emr-7.0.0
	m7gd.16xlarge	emr-5.36.1, emr-6.10.0, emr-7.0.0
Komputasi Dioptimalkan	c5.xlarge	emr-5.13.0, emr-6.0.0, emr-7.0.0
	c5.2xlarge	emr-5.13.0, emr-6.0.0, emr-7.0.0
	c5.4xlarge	emr-5.13.0, emr-6.0.0, emr-7.0.0
	c5.9xlarge	emr-5.13.0, emr-6.0.0, emr-7.0.0
	c5.12xlarge	emr-5.13.0, emr-6.0.0, emr-7.0.0
	c5.18xlarge	emr-5.13.0, emr-6.0.0, emr-7.0.0
	c5.24xlarge	emr-5.13.0, emr-6.0.0, emr-7.0.0
	c5a.xlarge	emr-5.31.0, emr-6.1.0, emr-7.0.0

Kelas Instans	Tipe instans	Versi EMR Amazon yang didukung minimum (5.x, 6.x, 7.x)
	c5a.2xlarge	emr-5.31.0, emr-6.1.0, emr-7.0.0
	c5a.4xlarge	emr-5.31.0, emr-6.1.0, emr-7.0.0
	c5a.8xlarge	emr-5.31.0, emr-6.1.0, emr-7.0.0
	c5a.12xlarge	emr-5.31.0, emr-6.1.0, emr-7.0.0
	c5a.16xlarge	emr-5.31.0, emr-6.1.0, emr-7.0.0
	c5a.24xlarge	emr-5.31.0, emr-6.1.0, emr-7.0.0
	c5ad.xlarge	emr-5.33.0, emr-6.3.0, emr-7.0.0
	c5ad.2xlarge	emr-5.33.0, emr-6.3.0, emr-7.0.0
	c5ad.4xlarge	emr-5.33.0, emr-6.3.0, emr-7.0.0
	c5ad.8xlarge	emr-5.33.0, emr-6.3.0, emr-7.0.0
	c5ad.12xlarge	emr-5.33.0, emr-6.3.0, emr-7.0.0
	c5ad.16xlarge	emr-5.33.0, emr-6.3.0, emr-7.0.0

Kelas Instans	Tipe instans	Versi EMR Amazon yang didukung minimum (5.x, 6.x, 7.x)
	c5ad.24xlarge	emr-5.33.0, emr-6.3.0, emr-7.0.0
	c5d.xlarge	emr-5.13.0, emr-6.0.0, emr-7.0.0
	c5d.2xlarge	emr-5.13.0, emr-6.0.0, emr-7.0.0
	c5d.4xlarge	emr-5.13.0, emr-6.0.0, emr-7.0.0
	c5d.9xlarge	emr-5.13.0, emr-6.0.0, emr-7.0.0
	c5d.12xlarge	emr-5.13.0, emr-6.0.0, emr-7.0.0
	c5d.18xlarge	emr-5.13.0, emr-6.0.0, emr-7.0.0
	c5d.24xlarge	emr-5.13.0, emr-6.0.0, emr-7.0.0
	c5n.xlarge	emr-5.20.0, emr-6.0.0, emr-7.0.0
	c5n.2xlarge	emr-5.20.0, emr-6.0.0, emr-7.0.0
	c5n.4xlarge	emr-5.20.0, emr-6.0.0, emr-7.0.0
	c5n.9xlarge	emr-5.20.0, emr-6.0.0, emr-7.0.0

Kelas Instans	Tipe instans	Versi EMR Amazon yang didukung minimum (5.x, 6.x, 7.x)
	c5n.18xlarge	emr-5.20.0, emr-6.0.0, emr-7.0.0
	c6a.xlarge	emr-5.36.1, emr-6.8.0, emr-7.0.0
	c6a.2xlarge	emr-5.36.1, emr-6.8.0, emr-7.0.0
	c6a.4xlarge	emr-5.36.1, emr-6.8.0, emr-7.0.0
	c6a.8xlarge	emr-5.36.1, emr-6.8.0, emr-7.0.0
	c6a.12xlarge	emr-5.36.1, emr-6.8.0, emr-7.0.0
	c6a.16xlarge	emr-5.36.1, emr-6.8.0, emr-7.0.0
	c6a.24xlarge	emr-5.36.1, emr-6.8.0, emr-7.0.0
	c6a.32xlarge	emr-5.36.1, emr-6.8.0, emr-7.0.0
	c6a.48xlarge	emr-5.36.1, emr-6.8.0, emr-7.0.0
	c6g.xlarge	emr-5.31.0, emr-6.1.0, emr-7.0.0
	c6g.2xlarge	emr-5.31.0, emr-6.1.0, emr-7.0.0

Kelas Instans	Tipe instans	Versi EMR Amazon yang didukung minimum (5.x, 6.x, 7.x)
	c6g.4xlarge	emr-5.31.0, emr-6.1.0, emr-7.0.0
	c6g.8xlarge	emr-5.31.0, emr-6.1.0, emr-7.0.0
	c6g.12xlarge	emr-5.31.0, emr-6.1.0, emr-7.0.0
	c6g.16xlarge	emr-5.31.0, emr-6.1.0, emr-7.0.0
	c6gd.xlarge	emr-5.33.0, emr-6.3.0, emr-7.0.0
	c6gd.2xlarge	emr-5.33.0, emr-6.3.0, emr-7.0.0
	c6gd.4xlarge	emr-5.33.0, emr-6.3.0, emr-7.0.0
	c6gd.8xlarge	emr-5.33.0, emr-6.3.0, emr-7.0.0
	c6gd.12xlarge	emr-5.33.0, emr-6.3.0, emr-7.0.0
	c6gd.16xlarge	emr-5.33.0, emr-6.3.0, emr-7.0.0
	c6gn.xlarge	emr-5.33.0, emr-6.3.0, emr-7.0.0
	c6gn.2xlarge	emr-5.33.0, emr-6.3.0, emr-7.0.0

Kelas Instans	Tipe instans	Versi EMR Amazon yang didukung minimum (5.x, 6.x, 7.x)
	c6gn.4xlarge	emr-5.33.0, emr-6.3.0, emr-7.0.0
	c6gn.8xlarge	emr-5.33.0, emr-6.3.0, emr-7.0.0
	c6gn.12xlarge	emr-5.33.0, emr-6.3.0, emr-7.0.0
	c6gn.16xlarge	emr-5.33.0, emr-6.3.0, emr-7.0.0
	c6i.xlarge	emr-5.35.0, emr-6.6.0
	c6i.2xlarge	emr-5.35.0, emr-6.6.0
	c6i.4xlarge	emr-5.35.0, emr-6.6.0
	c6i.8xlarge	emr-5.35.0, emr-6.6.0
	c6i.12xlarge	emr-5.35.0, emr-6.6.0
	c6i.16xlarge	emr-5.35.0, emr-6.6.0
	c6i.24xlarge	emr-5.35.0, emr-6.6.0
	c6i.32xlarge	emr-5.35.0, emr-6.6.0
	c6in.xlarge	emr-5.36.1, emr-6.10.0
	c6in.2xlarge	emr-5.36.1, emr-6.10.0
	c6in.4xlarge	emr-5.36.1, emr-6.10.0
	c6in.8xlarge	emr-5.36.1, emr-6.10.0
	c6in.12xlarge	emr-5.36.1, emr-6.10.0

Kelas Instans	Tipe instans	Versi EMR Amazon yang didukung minimum (5.x, 6.x, 7.x)
	c6in.16xlarge	emr-5.36.1, emr-6.10.0
	c6in.24xlarge	emr-5.36.1, emr-6.10.0
	c6in.32xlarge	emr-5.36.1, emr-6.10.0
	c7g.xlarge	emr-5.36.1, emr-6.7.0, emr-7.0.0
	c7g.2xlarge	emr-5.36.1, emr-6.7.0, emr-7.0.0
	c7g.4xlarge	emr-5.36.1, emr-6.7.0, emr-7.0.0
	c7g.8xlarge	emr-5.36.1, emr-6.7.0, emr-7.0.0
	c7g.12xlarge	emr-5.36.1, emr-6.7.0, emr-7.0.0
	c7g.16xlarge	emr-5.36.1, emr-6.7.0, emr-7.0.0
	c7gd.xlarge	emr-5.36.1, emr-6.10.0, emr-7.0.0
	c7gd.2xlarge	emr-5.36.1, emr-6.10.0, emr-7.0.0
	c7gd.4xlarge	emr-5.36.1, emr-6.10.0, emr-7.0.0
	c7gd.8xlarge	emr-5.36.1, emr-6.10.0, emr-7.0.0

Kelas Instans	Tipe instans	Versi EMR Amazon yang didukung minimum (5.x, 6.x, 7.x)
	c7gd.12xlarge	emr-5.36.1, emr-6.10.0, emr-7.0.0
	c7gd.16xlarge	emr-5.36.1, emr-6.10.0, emr-7.0.0
Komputasi yang Dipercepat	g3.4xlarge	emr-5.18.0, emr-6.0.0, emr-7.0.0
	g3.8xlarge	emr-5.18.0, emr-6.0.0, emr-7.0.0
	g3.16xlarge	emr-5.18.0, emr-6.0.0, emr-7.0.0
	g4dn.xlarge	emr-5.30.0, emr-6.0.0, emr-7.0.0
	g4dn.2xlarge	emr-5.30.0, emr-6.0.0, emr-7.0.0
	g4dn.4xlarge	emr-5.30.0, emr-6.0.0, emr-7.0.0
	g4dn.8xlarge	emr-5.30.0, emr-6.0.0, emr-7.0.0
	g4dn.12xlarge	emr-5.30.0, emr-6.0.0, emr-7.0.0
	g4dn.16xlarge	emr-5.30.0, emr-6.0.0, emr-7.0.0
	p2.xlarge	emr-5.10.0, emr-6.0.0, emr-7.0.0

Kelas Instans	Tipe instans	Versi EMR Amazon yang didukung minimum (5.x, 6.x, 7.x)
	p2.8xlarge	emr-5.10.0, emr-6.0.0, emr-7.0.0
	p2.16xlarge	emr-5.10.0, emr-6.0.0, emr-7.0.0
	p3.2xlarge	emr-5.10.0, emr-6.0.0, emr-7.0.0
	p3.8xlarge	emr-5.10.0, emr-6.0.0, emr-7.0.0
	p3.16xlarge	emr-5.10.0, emr-6.0.0, emr-7.0.0
Memori Dioptimalkan	r5.xlarge	emr-5.13.0, emr-6.0.0, emr-7.0.0
	r5.2xlarge	emr-5.13.0, emr-6.0.0, emr-7.0.0
	r5.4xlarge	emr-5.13.0, emr-6.0.0, emr-7.0.0
	r5.8xlarge	emr-5.13.0, emr-6.0.0, emr-7.0.0
	r5.12xlarge	emr-5.13.0, emr-6.0.0, emr-7.0.0
	r5.16xlarge	emr-5.13.0, emr-6.0.0, emr-7.0.0
	r5.24xlarge	emr-5.13.0, emr-6.0.0, emr-7.0.0

Kelas Instans	Tipe instans	Versi EMR Amazon yang didukung minimum (5.x, 6.x, 7.x)
	r5a.xlarge	emr-5.20.0, emr-6.0.0, emr-7.0.0
	r5a.2xlarge	emr-5.20.0, emr-6.0.0, emr-7.0.0
	r5a.4xlarge	emr-5.20.0, emr-6.0.0, emr-7.0.0
	r5a.8xlarge	emr-5.20.0, emr-6.0.0, emr-7.0.0
	r5a.12xlarge	emr-5.20.0, emr-6.0.0, emr-7.0.0
	r5a.16xlarge	emr-5.20.0, emr-6.0.0, emr-7.0.0
	r5a.24xlarge	emr-5.20.0, emr-6.0.0, emr-7.0.0
	r5ad.xlarge	emr-5.20.0, emr-6.0.0, emr-7.0.0
	r5ad.2xlarge	emr-5.20.0, emr-6.0.0, emr-7.0.0
	r5ad.4xlarge	emr-5.20.0, emr-6.0.0, emr-7.0.0
	r5ad.8xlarge	emr-5.33.0, emr-6.3.0, emr-7.0.0
	r5ad.12xlarge	emr-5.20.0, emr-6.0.0, emr-7.0.0

Kelas Instans	Tipe instans	Versi EMR Amazon yang didukung minimum (5.x, 6.x, 7.x)
	r5ad.16xlarge	emr-5.33.0, emr-6.3.0, emr-7.0.0
	r5ad.24xlarge	emr-5.20.0, emr-6.0.0, emr-7.0.0
	r5b.xlarge	emr-5.33.0, emr-6.3.0, emr-7.0.0
	r5b.2xlarge	emr-5.33.0, emr-6.3.0, emr-7.0.0
	r5b.4xlarge	emr-5.33.0, emr-6.3.0, emr-7.0.0
	r5b.8xlarge	emr-5.33.0, emr-6.3.0, emr-7.0.0
	r5b.12xlarge	emr-5.33.0, emr-6.3.0, emr-7.0.0
	r5b.16xlarge	emr-5.33.0, emr-6.3.0, emr-7.0.0
	r5b.24xlarge	emr-5.33.0, emr-6.3.0, emr-7.0.0
	r5d.xlarge	emr-5.13.0, emr-6.0.0, emr-7.0.0
	r5d.2xlarge	emr-5.13.0, emr-6.0.0, emr-7.0.0
	r5d.4xlarge	emr-5.13.0, emr-6.0.0, emr-7.0.0

Kelas Instans	Tipe instans	Versi EMR Amazon yang didukung minimum (5.x, 6.x, 7.x)
	r5d.8xlarge	emr-5.13.0, emr-6.0.0, emr-7.0.0
	r5d.12xlarge	emr-5.13.0, emr-6.0.0, emr-7.0.0
	r5d.16xlarge	emr-5.13.0, emr-6.0.0, emr-7.0.0
	r5d.24xlarge	emr-5.13.0, emr-6.0.0, emr-7.0.0
	r5dn.xlarge	emr-5.31.0, emr-6.1.0, emr-7.0.0
	r5dn.2xlarge	emr-5.31.0, emr-6.1.0, emr-7.0.0
	r5dn.4xlarge	emr-5.31.0, emr-6.1.0, emr-7.0.0
	r5dn.8xlarge	emr-5.31.0, emr-6.1.0, emr-7.0.0
	r5dn.12xlarge	emr-5.31.0, emr-6.1.0, emr-7.0.0
	r5dn.16xlarge	emr-5.31.0, emr-6.1.0, emr-7.0.0
	r5dn.24xlarge	emr-5.31.0, emr-6.1.0, emr-7.0.0
	r5n.xlarge	emr-5.31.0, emr-6.1.0, emr-7.0.0

Kelas Instans	Tipe instans	Versi EMR Amazon yang didukung minimum (5.x, 6.x, 7.x)
	r5n.2xlarge	emr-5.31.0, emr-6.1.0, emr-7.0.0
	r5n.4xlarge	emr-5.31.0, emr-6.1.0, emr-7.0.0
	r5n.8xlarge	emr-5.31.0, emr-6.1.0, emr-7.0.0
	r5n.12xlarge	emr-5.31.0, emr-6.1.0, emr-7.0.0
	r5n.16xlarge	emr-5.31.0, emr-6.1.0, emr-7.0.0
	r5n.24xlarge	emr-5.31.0, emr-6.1.0, emr-7.0.0
	r6a.xlarge	emr-5.36.1, emr-6.8.0, emr-7.0.0
	r6a.2xlarge	emr-5.36.1, emr-6.8.0, emr-7.0.0
	r6a.4xlarge	emr-5.36.1, emr-6.8.0, emr-7.0.0
	r6a.8xlarge	emr-5.36.1, emr-6.8.0, emr-7.0.0
	r6a.12xlarge	emr-5.36.1, emr-6.8.0, emr-7.0.0
	r6a.16xlarge	emr-5.36.1, emr-6.8.0, emr-7.0.0

Kelas Instans	Tipe instans	Versi EMR Amazon yang didukung minimum (5.x, 6.x, 7.x)
	r6a.24xlarge	emr-5.36.1, emr-6.8.0, emr-7.0.0
	r6a.32xlarge	emr-5.36.1, emr-6.8.0, emr-7.0.0
	r6a.48xlarge	emr-5.36.1, emr-6.8.0, emr-7.0.0
	r6g.xlarge	emr-5.31.0, emr-6.1.0, emr-7.0.0
	r6g.2xlarge	emr-5.31.0, emr-6.1.0, emr-7.0.0
	r6g.4xlarge	emr-5.31.0, emr-6.1.0, emr-7.0.0
	r6g.8xlarge	emr-5.31.0, emr-6.1.0, emr-7.0.0
	r6g.12xlarge	emr-5.31.0, emr-6.1.0, emr-7.0.0
	r6g.16xlarge	emr-5.31.0, emr-6.1.0, emr-7.0.0
	r6gd.xlarge	emr-5.33.0, emr-6.3.0, emr-7.0.0
	r6gd.2xlarge	emr-5.33.0, emr-6.3.0, emr-7.0.0
	r6gd.4xlarge	emr-5.33.0, emr-6.3.0, emr-7.0.0

Kelas Instans	Tipe instans	Versi EMR Amazon yang didukung minimum (5.x, 6.x, 7.x)
	r6gd.8xlarge	emr-5.33.0, emr-6.3.0, emr-7.0.0
	r6gd.12xlarge	emr-5.33.0, emr-6.3.0, emr-7.0.0
	r6gd.16xlarge	emr-5.33.0, emr-6.3.0, emr-7.0.0
	r6i.xlarge	emr-5.35.0, emr-6.6.0
	r6i.2xlarge	emr-5.35.0, emr-6.6.0
	r6i.4xlarge	emr-5.35.0, emr-6.6.0
	r6i.8xlarge	emr-5.35.0, emr-6.6.0
	r6i.12xlarge	emr-5.35.0, emr-6.6.0
	r6i.16xlarge	emr-5.35.0, emr-6.6.0
	r6i.24xlarge	emr-5.35.0, emr-6.6.0
	r6i.32xlarge	emr-5.35.0, emr-6.6.0
	r6id.xlarge	emr-5.36.1, emr-6.8.0
	r6id.2xlarge	emr-5.36.1, emr-6.8.0
	r6id.4xlarge	emr-5.36.1, emr-6.8.0
	r6id.8xlarge	emr-5.36.1, emr-6.8.0
	r6id.12xlarge	emr-5.36.1, emr-6.8.0
	r6id.16xlarge	emr-5.36.1, emr-6.8.0

Kelas Instans	Tipe instans	Versi EMR Amazon yang didukung minimum (5.x, 6.x, 7.x)
	r6id.24xlarge	emr-5.36.1, emr-6.8.0
	r6id.32xlarge	emr-5.36.1, emr-6.8.0
	r6idn.xlarge	emr-5.36.1, emr-6.10.0
	r6idn.2xlarge	emr-5.36.1, emr-6.10.0
	r6idn.4xlarge	emr-5.36.1, emr-6.10.0
	r6idn.8xlarge	emr-5.36.1, emr-6.10.0
	r6idn.12xlarge	emr-5.36.1, emr-6.10.0
	r6idn.16xlarge	emr-5.36.1, emr-6.10.0
	r6idn.24xlarge	emr-5.36.1, emr-6.10.0
	r6idn.32xlarge	emr-5.36.1, emr-6.10.0
	r6in.xlarge	emr-5.36.1, emr-6.10.0
	r6in.2xlarge	emr-5.36.1, emr-6.10.0
	r6in.4xlarge	emr-5.36.1, emr-6.10.0
	r6in.8xlarge	emr-5.36.1, emr-6.10.0
	r6in.12xlarge	emr-5.36.1, emr-6.10.0
	r6in.16xlarge	emr-5.36.1, emr-6.10.0
	r6in.24xlarge	emr-5.36.1, emr-6.10.0
	r6in.32xlarge	emr-5.36.1, emr-6.10.0
	r7g.xlarge	emr-5.36.1, emr-6.10.0, emr-7.0.0

Kelas Instans	Tipe instans	Versi EMR Amazon yang didukung minimum (5.x, 6.x, 7.x)
	r7g.2xlarge	emr-5.36.1, emr-6.10.0, emr-7.0.0
	r7g.4xlarge	emr-5.36.1, emr-6.10.0, emr-7.0.0
	r7g.8xlarge	emr-5.36.1, emr-6.10.0, emr-7.0.0
	r7g.12xlarge	emr-5.36.1, emr-6.10.0, emr-7.0.0
	r7g.16xlarge	emr-5.36.1, emr-6.10.0, emr-7.0.0
	r7gd.xlarge	emr-5.36.1, emr-6.10.0, emr-7.0.0
	r7gd.2xlarge	emr-5.36.1, emr-6.10.0, emr-7.0.0
	r7gd.4xlarge	emr-5.36.1, emr-6.10.0, emr-7.0.0
	r7gd.8xlarge	emr-5.36.1, emr-6.10.0, emr-7.0.0
	r7gd.12xlarge	emr-5.36.1, emr-6.10.0, emr-7.0.0
	r7gd.16xlarge	emr-5.36.1, emr-6.10.0, emr-7.0.0
	x1.16xlarge	emr-5.36.1, emr-6.9.0, emr-7.0.0

Kelas Instans	Tipe instans	Versi EMR Amazon yang didukung minimum (5.x, 6.x, 7.x)
	x1.32xlarge	emr-5.36.1, emr-6.9.0, emr-7.0.0
	x1e.xlarge	emr-5.36.1, emr-6.10.0, emr-7.0.0
	x1e.2xlarge	emr-5.36.1, emr-6.10.0, emr-7.0.0
	x1e.4xlarge	emr-5.36.1, emr-6.10.0, emr-7.0.0
	x1e.8xlarge	emr-5.36.1, emr-6.10.0, emr-7.0.0
	x1e.16xlarge	emr-5.36.1, emr-6.10.0, emr-7.0.0
	x1e.32xlarge	emr-5.36.1, emr-6.10.0, emr-7.0.0
	x2idn.16xlarge	emr-5.36.1, emr-6.9.0
	x2idn.24xlarge	emr-5.36.1, emr-6.9.0
	x2idn.32xlarge	emr-5.36.1, emr-6.9.0
	x2iedn.xlarge	emr-5.36.1, emr-6.9.0
	x2iedn.2xlarge	emr-5.36.1, emr-6.9.0
	x2iedn.4xlarge	emr-5.36.1, emr-6.9.0
	x2iedn.8xlarge	emr-5.36.1, emr-6.9.0
	x2iedn.16xlarge	emr-5.36.1, emr-6.9.0

Kelas Instans	Tipe instans	Versi EMR Amazon yang didukung minimum (5.x, 6.x, 7.x)
	x2iedn.24xlarge	emr-5.36.1, emr-6.9.0
	x2iedn.32xlarge	emr-5.36.1, emr-6.9.0
	z1d.xlarge	emr-5.17.0, emr-6.0.0, emr-7.0.0
	z1d.2xlarge	emr-5.17.0, emr-6.0.0, emr-7.0.0
	z1d.3xlarge	emr-5.17.0, emr-6.0.0, emr-7.0.0
	z1d.6xlarge	emr-5.17.0, emr-6.0.0, emr-7.0.0
	z1d.12xlarge	emr-5.17.0, emr-6.0.0, emr-7.0.0
Penyimpanan Dioptimalkan	d3.xlarge	emr-5.33.0, emr-6.3.0, emr-7.0.0
	d3.2xlarge	emr-5.33.0, emr-6.3.0, emr-7.0.0
	d3.4xlarge	emr-5.33.0, emr-6.3.0, emr-7.0.0
	d3.8xlarge	emr-5.33.0, emr-6.3.0, emr-7.0.0
	d3en.xlarge	emr-5.33.0, emr-6.3.0, emr-7.0.0
	d3en.2xlarge	emr-5.33.0, emr-6.3.0, emr-7.0.0

Kelas Instans	Tipe instans	Versi EMR Amazon yang didukung minimum (5.x, 6.x, 7.x)
	d3en.4xlarge	emr-5.33.0, emr-6.3.0, emr-7.0.0
	d3en.6xlarge	emr-5.33.0, emr-6.3.0, emr-7.0.0
	d3en.8xlarge	emr-5.33.0, emr-6.3.0, emr-7.0.0
	d3en.12xlarge	emr-5.33.0, emr-6.3.0, emr-7.0.0
	i3.xlarge	emr-5.9.0, emr-6.0.0, emr-7.0.0
	i3.2xlarge	emr-5.9.0, emr-6.0.0, emr-7.0.0
	i3.4xlarge	emr-5.9.0, emr-6.0.0, emr-7.0.0
	i3.8xlarge	emr-5.9.0, emr-6.0.0, emr-7.0.0
	i3.16xlarge	emr-5.9.0, emr-6.0.0, emr-7.0.0
	i3en.xlarge	emr-5.25.0, emr-6.0.0, emr-7.0.0
	i3en.2xlarge	emr-5.25.0, emr-6.0.0, emr-7.0.0
	i3en.3xlarge	emr-5.25.0, emr-6.0.0, emr-7.0.0

Kelas Instans	Tipe instans	Versi EMR Amazon yang didukung minimum (5.x, 6.x, 7.x)
	i3en.6xlarge	emr-5.25.0, emr-6.0.0, emr-7.0.0
	i3en.12xlarge	emr-5.25.0, emr-6.0.0, emr-7.0.0
	i3en.24xlarge	emr-5.25.0, emr-6.0.0, emr-7.0.0
	i4i.xlarge	emr-5.36.1, emr-6.8.0
	i4i.2xlarge	emr-5.36.1, emr-6.8.0
	i4i.4xlarge	emr-5.36.1, emr-6.8.0
	i4i.8xlarge	emr-5.36.1, emr-6.8.0
	i4i.12xlarge	emr-5.36.1, emr-6.10.0
	i4i.16xlarge	emr-5.36.1, emr-6.8.0
	i4i.24xlarge	emr-5.36.1, emr-6.10.0
	i4i.32xlarge	emr-5.36.1, emr-6.8.0
	im4gn.xlarge	emr-5.35.0, emr-6.6.0, emr-7.0.0
	im4gn.2xlarge	emr-5.35.0, emr-6.6.0, emr-7.0.0
	im4gn.4xlarge	emr-5.35.0, emr-6.6.0, emr-7.0.0
	im4gn.8xlarge	emr-5.35.0, emr-6.6.0, emr-7.0.0

Kelas Instans	Tipe instans	Versi EMR Amazon yang didukung minimum (5.x, 6.x, 7.x)
	im4gn.16xlarge	emr-5.35.0, emr-6.6.0, emr-7.0.0
	is4gen.xlarge	emr-5.35.0, emr-6.6.0, emr-7.0.0
	is4gen.2xlarge	emr-5.35.0, emr-6.6.0, emr-7.0.0
	is4gen.4xlarge	emr-5.35.0, emr-6.6.0, emr-7.0.0
	is4gen.8xlarge	emr-5.35.0, emr-6.6.0, emr-7.0.0

Asia Pacific (Sydney) - ap-southeast-2

Kelas Instans	Tipe instans	Versi EMR Amazon yang didukung minimum (5.x, 6.x, 7.x)
Tujuan Umum	m5.xlarge	emr-5.13.0, emr-6.0.0
	m5.2xlarge	emr-5.13.0, emr-6.0.0
	m5.4xlarge	emr-5.13.0, emr-6.0.0
	m5.8xlarge	emr-5.13.0, emr-6.0.0
	m5.12xlarge	emr-5.13.0, emr-6.0.0
	m5.16xlarge	emr-5.13.0, emr-6.0.0
	m5.24xlarge	emr-5.13.0, emr-6.0.0

Kelas Instans	Tipe instans	Versi EMR Amazon yang didukung minimum (5.x, 6.x, 7.x)
	m5a.xlarge	emr-5.20.0, emr-6.0.0
	m5a.2xlarge	emr-5.20.0, emr-6.0.0
	m5a.4xlarge	emr-5.20.0, emr-6.0.0
	m5a.8xlarge	emr-5.20.0, emr-6.0.0
	m5a.12xlarge	emr-5.20.0, emr-6.0.0
	m5a.16xlarge	emr-5.20.0, emr-6.0.0
	m5a.24xlarge	emr-5.20.0, emr-6.0.0
	m5ad.xlarge	emr-5.20.0, emr-6.0.0
	m5ad.2xlarge	emr-5.20.0, emr-6.0.0
	m5ad.4xlarge	emr-5.20.0, emr-6.0.0
	m5ad.8xlarge	emr-5.33.0, emr-6.3.0
	m5ad.12xlarge	emr-5.20.0, emr-6.0.0
	m5ad.16xlarge	emr-5.33.0, emr-6.3.0
	m5ad.24xlarge	emr-5.20.0, emr-6.0.0
	m5d.xlarge	emr-5.13.0, emr-6.0.0
	m5d.2xlarge	emr-5.13.0, emr-6.0.0
	m5d.4xlarge	emr-5.13.0, emr-6.0.0
	m5d.8xlarge	emr-5.13.0, emr-6.0.0
	m5d.12xlarge	emr-5.13.0, emr-6.0.0

Kelas Instans	Tipe instans	Versi EMR Amazon yang didukung minimum (5.x, 6.x, 7.x)
	m5d.16xlarge	emr-5.13.0, emr-6.0.0
	m5d.24xlarge	emr-5.13.0, emr-6.0.0
	m5zn.xlarge	emr-5.33.0, emr-6.3.0
	m5zn.2xlarge	emr-5.33.0, emr-6.3.0
	m5zn.3xlarge	emr-5.33.0, emr-6.3.0
	m5zn.6xlarge	emr-5.33.0, emr-6.3.0
	m5zn.12xlarge	emr-5.33.0, emr-6.3.0
	m6a.xlarge	emr-5.36.1, emr-6.8.0
	m6a.2xlarge	emr-5.36.1, emr-6.8.0
	m6a.4xlarge	emr-5.36.1, emr-6.8.0
	m6a.8xlarge	emr-5.36.1, emr-6.8.0
	m6a.12xlarge	emr-5.36.1, emr-6.8.0
	m6a.16xlarge	emr-5.36.1, emr-6.8.0
	m6a.24xlarge	emr-5.36.1, emr-6.8.0
	m6a.32xlarge	emr-5.36.1, emr-6.8.0
	m6a.48xlarge	emr-5.36.1, emr-6.8.0
	m6g.xlarge	emr-5.30.0, emr-6.1.0
	m6g.2xlarge	emr-5.30.0, emr-6.1.0
	m6g.4xlarge	emr-5.30.0, emr-6.1.0

Kelas Instans	Tipe instans	Versi EMR Amazon yang didukung minimum (5.x, 6.x, 7.x)
	m6g.8xlarge	emr-5.30.0, emr-6.1.0
	m6g.12xlarge	emr-5.30.0, emr-6.1.0
	m6g.16xlarge	emr-5.30.0, emr-6.1.0
	m6gd.xlarge	emr-5.33.0, emr-6.3.0
	m6gd.2xlarge	emr-5.33.0, emr-6.3.0
	m6gd.4xlarge	emr-5.33.0, emr-6.3.0
	m6gd.8xlarge	emr-5.33.0, emr-6.3.0
	m6gd.12xlarge	emr-5.33.0, emr-6.3.0
	m6gd.16xlarge	emr-5.33.0, emr-6.3.0
	m6i.xlarge	emr-5.35.0, emr-6.6.0
	m6i.2xlarge	emr-5.35.0, emr-6.6.0
	m6i.4xlarge	emr-5.35.0, emr-6.6.0
	m6i.8xlarge	emr-5.35.0, emr-6.6.0
	m6i.12xlarge	emr-5.35.0, emr-6.6.0
	m6i.16xlarge	emr-5.35.0, emr-6.6.0
	m6i.24xlarge	emr-5.35.0, emr-6.6.0
	m6i.32xlarge	emr-5.35.0, emr-6.6.0
	m6id.xlarge	emr-5.36.1, emr-6.8.0
	m6id.2xlarge	emr-5.36.1, emr-6.8.0

Kelas Instans	Tipe instans	Versi EMR Amazon yang didukung minimum (5.x, 6.x, 7.x)
	m6id.4xlarge	emr-5.36.1, emr-6.8.0
	m6id.8xlarge	emr-5.36.1, emr-6.8.0
	m6id.12xlarge	emr-5.36.1, emr-6.8.0
	m6id.16xlarge	emr-5.36.1, emr-6.8.0
	m6id.24xlarge	emr-5.36.1, emr-6.8.0
	m6id.32xlarge	emr-5.36.1, emr-6.8.0
	m6idn.xlarge	emr-5.36.1, emr-6.10.0
	m6idn.2xlarge	emr-5.36.1, emr-6.10.0
	m6idn.4xlarge	emr-5.36.1, emr-6.10.0
	m6idn.8xlarge	emr-5.36.1, emr-6.10.0
	m6idn.12xlarge	emr-5.36.1, emr-6.10.0
	m6idn.16xlarge	emr-5.36.1, emr-6.10.0
	m6idn.24xlarge	emr-5.36.1, emr-6.10.0
	m6idn.32xlarge	emr-5.36.1, emr-6.10.0
	m6in.xlarge	emr-5.36.1, emr-6.10.0
	m6in.2xlarge	emr-5.36.1, emr-6.10.0
	m6in.4xlarge	emr-5.36.1, emr-6.10.0
	m6in.8xlarge	emr-5.36.1, emr-6.10.0
	m6in.12xlarge	emr-5.36.1, emr-6.10.0

Kelas Instans	Tipe instans	Versi EMR Amazon yang didukung minimum (5.x, 6.x, 7.x)
	m6in.16xlarge	emr-5.36.1, emr-6.10.0
	m6in.24xlarge	emr-5.36.1, emr-6.10.0
	m6in.32xlarge	emr-5.36.1, emr-6.10.0
	m7g.xlarge	emr-5.36.1, emr-6.10.0
	m7g.2xlarge	emr-5.36.1, emr-6.10.0
	m7g.4xlarge	emr-5.36.1, emr-6.10.0
	m7g.8xlarge	emr-5.36.1, emr-6.10.0
	m7g.12xlarge	emr-5.36.1, emr-6.10.0
	m7g.16xlarge	emr-5.36.1, emr-6.10.0
	m7gd.xlarge	emr-5.36.1, emr-6.10.0
	m7gd.2xlarge	emr-5.36.1, emr-6.10.0
	m7gd.4xlarge	emr-5.36.1, emr-6.10.0
	m7gd.8xlarge	emr-5.36.1, emr-6.10.0
	m7gd.12xlarge	emr-5.36.1, emr-6.10.0
	m7gd.16xlarge	emr-5.36.1, emr-6.10.0
Komputasi Dioptimalkan	c5.xlarge	emr-5.13.0, emr-6.0.0
	c5.2xlarge	emr-5.13.0, emr-6.0.0
	c5.4xlarge	emr-5.13.0, emr-6.0.0
	c5.9xlarge	emr-5.13.0, emr-6.0.0

Kelas Instans	Tipe instans	Versi EMR Amazon yang didukung minimum (5.x, 6.x, 7.x)
	c5.12xlarge	emr-5.13.0, emr-6.0.0
	c5.18xlarge	emr-5.13.0, emr-6.0.0
	c5.24xlarge	emr-5.13.0, emr-6.0.0
	c5a.xlarge	emr-5.31.0, emr-6.1.0
	c5a.2xlarge	emr-5.31.0, emr-6.1.0
	c5a.4xlarge	emr-5.31.0, emr-6.1.0
	c5a.8xlarge	emr-5.31.0, emr-6.1.0
	c5a.12xlarge	emr-5.31.0, emr-6.1.0
	c5a.16xlarge	emr-5.31.0, emr-6.1.0
	c5a.24xlarge	emr-5.31.0, emr-6.1.0
	c5ad.xlarge	emr-5.33.0, emr-6.3.0
	c5ad.2xlarge	emr-5.33.0, emr-6.3.0
	c5ad.4xlarge	emr-5.33.0, emr-6.3.0
	c5ad.8xlarge	emr-5.33.0, emr-6.3.0
	c5ad.12xlarge	emr-5.33.0, emr-6.3.0
	c5ad.16xlarge	emr-5.33.0, emr-6.3.0
	c5ad.24xlarge	emr-5.33.0, emr-6.3.0
	c5d.xlarge	emr-5.13.0, emr-6.0.0
	c5d.2xlarge	emr-5.13.0, emr-6.0.0

Kelas Instans	Tipe instans	Versi EMR Amazon yang didukung minimum (5.x, 6.x, 7.x)
	c5d.4xlarge	emr-5.13.0, emr-6.0.0
	c5d.9xlarge	emr-5.13.0, emr-6.0.0
	c5d.12xlarge	emr-5.13.0, emr-6.0.0
	c5d.18xlarge	emr-5.13.0, emr-6.0.0
	c5d.24xlarge	emr-5.13.0, emr-6.0.0
	c5n.xlarge	emr-5.20.0, emr-6.0.0
	c5n.2xlarge	emr-5.20.0, emr-6.0.0
	c5n.4xlarge	emr-5.20.0, emr-6.0.0
	c5n.9xlarge	emr-5.20.0, emr-6.0.0
	c5n.18xlarge	emr-5.20.0, emr-6.0.0
	c6a.xlarge	emr-5.36.1, emr-6.8.0
	c6a.2xlarge	emr-5.36.1, emr-6.8.0
	c6a.4xlarge	emr-5.36.1, emr-6.8.0
	c6a.8xlarge	emr-5.36.1, emr-6.8.0
	c6a.12xlarge	emr-5.36.1, emr-6.8.0
	c6a.16xlarge	emr-5.36.1, emr-6.8.0
	c6a.24xlarge	emr-5.36.1, emr-6.8.0
	c6a.32xlarge	emr-5.36.1, emr-6.8.0
	c6a.48xlarge	emr-5.36.1, emr-6.8.0

Kelas Instans	Tipe instans	Versi EMR Amazon yang didukung minimum (5.x, 6.x, 7.x)
	c6g.xlarge	emr-5.31.0, emr-6.1.0
	c6g.2xlarge	emr-5.31.0, emr-6.1.0
	c6g.4xlarge	emr-5.31.0, emr-6.1.0
	c6g.8xlarge	emr-5.31.0, emr-6.1.0
	c6g.12xlarge	emr-5.31.0, emr-6.1.0
	c6g.16xlarge	emr-5.31.0, emr-6.1.0
	c6gd.xlarge	emr-5.33.0, emr-6.3.0
	c6gd.2xlarge	emr-5.33.0, emr-6.3.0
	c6gd.4xlarge	emr-5.33.0, emr-6.3.0
	c6gd.8xlarge	emr-5.33.0, emr-6.3.0
	c6gd.12xlarge	emr-5.33.0, emr-6.3.0
	c6gd.16xlarge	emr-5.33.0, emr-6.3.0
	c6gn.xlarge	emr-5.33.0, emr-6.3.0
	c6gn.2xlarge	emr-5.33.0, emr-6.3.0
	c6gn.4xlarge	emr-5.33.0, emr-6.3.0
	c6gn.8xlarge	emr-5.33.0, emr-6.3.0
	c6gn.12xlarge	emr-5.33.0, emr-6.3.0
	c6gn.16xlarge	emr-5.33.0, emr-6.3.0
	c6i.xlarge	emr-5.35.0, emr-6.6.0

Kelas Instans	Tipe instans	Versi EMR Amazon yang didukung minimum (5.x, 6.x, 7.x)
	c6i.2xlarge	emr-5.35.0, emr-6.6.0
	c6i.4xlarge	emr-5.35.0, emr-6.6.0
	c6i.8xlarge	emr-5.35.0, emr-6.6.0
	c6i.12xlarge	emr-5.35.0, emr-6.6.0
	c6i.16xlarge	emr-5.35.0, emr-6.6.0
	c6i.24xlarge	emr-5.35.0, emr-6.6.0
	c6i.32xlarge	emr-5.35.0, emr-6.6.0
	c6id.xlarge	emr-5.36.1, emr-6.8.0
	c6id.2xlarge	emr-5.36.1, emr-6.8.0
	c6id.4xlarge	emr-5.36.1, emr-6.8.0
	c6id.8xlarge	emr-5.36.1, emr-6.8.0
	c6id.12xlarge	emr-5.36.1, emr-6.8.0
	c6id.16xlarge	emr-5.36.1, emr-6.8.0
	c6id.24xlarge	emr-5.36.1, emr-6.8.0
	c6id.32xlarge	emr-5.36.1, emr-6.8.0
	c6in.xlarge	emr-5.36.1, emr-6.10.0
	c6in.2xlarge	emr-5.36.1, emr-6.10.0
	c6in.4xlarge	emr-5.36.1, emr-6.10.0
	c6in.8xlarge	emr-5.36.1, emr-6.10.0

Kelas Instans	Tipe instans	Versi EMR Amazon yang didukung minimum (5.x, 6.x, 7.x)
	c6in.12xlarge	emr-5.36.1, emr-6.10.0
	c6in.16xlarge	emr-5.36.1, emr-6.10.0
	c6in.24xlarge	emr-5.36.1, emr-6.10.0
	c6in.32xlarge	emr-5.36.1, emr-6.10.0
	c7g.xlarge	emr-5.36.1, emr-6.7.0
	c7g.2xlarge	emr-5.36.1, emr-6.7.0
	c7g.4xlarge	emr-5.36.1, emr-6.7.0
	c7g.8xlarge	emr-5.36.1, emr-6.7.0
	c7g.12xlarge	emr-5.36.1, emr-6.7.0
	c7g.16xlarge	emr-5.36.1, emr-6.7.0
	c7gd.xlarge	emr-5.36.1, emr-6.10.0
	c7gd.2xlarge	emr-5.36.1, emr-6.10.0
	c7gd.4xlarge	emr-5.36.1, emr-6.10.0
	c7gd.8xlarge	emr-5.36.1, emr-6.10.0
	c7gd.12xlarge	emr-5.36.1, emr-6.10.0
	c7gd.16xlarge	emr-5.36.1, emr-6.10.0
Komputasi yang Dipercepat	g3.4xlarge	emr-5.18.0, emr-6.0.0
	g3.8xlarge	emr-5.18.0, emr-6.0.0
	g3.16xlarge	emr-5.18.0, emr-6.0.0

Kelas Instans	Tipe instans	Versi EMR Amazon yang didukung minimum (5.x, 6.x, 7.x)
	g3s.xlarge	emr-5.19.0, emr-6.0.0
	g4dn.xlarge	emr-5.30.0, emr-6.0.0
	g4dn.2xlarge	emr-5.30.0, emr-6.0.0
	g4dn.4xlarge	emr-5.30.0, emr-6.0.0
	g4dn.8xlarge	emr-5.30.0, emr-6.0.0
	g4dn.12xlarge	emr-5.30.0, emr-6.0.0
	g4dn.16xlarge	emr-5.30.0, emr-6.0.0
	g5.xlarge	emr-5.36.1, emr-6.9.0
	g5.2xlarge	emr-5.36.1, emr-6.9.0
	g5.4xlarge	emr-5.36.1, emr-6.9.0
	g5.8xlarge	emr-5.36.1, emr-6.9.0
	g5.12xlarge	emr-5.36.1, emr-6.9.0
	g5.16xlarge	emr-5.36.1, emr-6.9.0
	g5.24xlarge	emr-5.36.1, emr-6.9.0
	g5.48xlarge	emr-5.36.1, emr-6.9.0
	p2.xlarge	emr-5.10.0, emr-6.0.0
	p2.8xlarge	emr-5.10.0, emr-6.0.0
	p2.16xlarge	emr-5.10.0, emr-6.0.0
	p3.2xlarge	emr-5.10.0, emr-6.0.0

Kelas Instans	Tipe instans	Versi EMR Amazon yang didukung minimum (5.x, 6.x, 7.x)
	p3.8xlarge	emr-5.10.0, emr-6.0.0
	p3.16xlarge	emr-5.10.0, emr-6.0.0
Memori Dioptimalkan	r5.xlarge	emr-5.13.0, emr-6.0.0
	r5.2xlarge	emr-5.13.0, emr-6.0.0
	r5.4xlarge	emr-5.13.0, emr-6.0.0
	r5.8xlarge	emr-5.13.0, emr-6.0.0
	r5.12xlarge	emr-5.13.0, emr-6.0.0
	r5.16xlarge	emr-5.13.0, emr-6.0.0
	r5.24xlarge	emr-5.13.0, emr-6.0.0
	r5a.xlarge	emr-5.20.0, emr-6.0.0
	r5a.2xlarge	emr-5.20.0, emr-6.0.0
	r5a.4xlarge	emr-5.20.0, emr-6.0.0
	r5a.8xlarge	emr-5.20.0, emr-6.0.0
	r5a.12xlarge	emr-5.20.0, emr-6.0.0
	r5a.16xlarge	emr-5.20.0, emr-6.0.0
	r5a.24xlarge	emr-5.20.0, emr-6.0.0
	r5ad.xlarge	emr-5.20.0, emr-6.0.0
	r5ad.2xlarge	emr-5.20.0, emr-6.0.0
	r5ad.4xlarge	emr-5.20.0, emr-6.0.0

Kelas Instans	Tipe instans	Versi EMR Amazon yang didukung minimum (5.x, 6.x, 7.x)
	r5ad.8xlarge	emr-5.33.0, emr-6.3.0
	r5ad.12xlarge	emr-5.20.0, emr-6.0.0
	r5ad.16xlarge	emr-5.33.0, emr-6.3.0
	r5ad.24xlarge	emr-5.20.0, emr-6.0.0
	r5b.xlarge	emr-5.33.0, emr-6.3.0
	r5b.2xlarge	emr-5.33.0, emr-6.3.0
	r5b.4xlarge	emr-5.33.0, emr-6.3.0
	r5b.8xlarge	emr-5.33.0, emr-6.3.0
	r5b.12xlarge	emr-5.33.0, emr-6.3.0
	r5b.16xlarge	emr-5.33.0, emr-6.3.0
	r5b.24xlarge	emr-5.33.0, emr-6.3.0
	r5d.xlarge	emr-5.13.0, emr-6.0.0
	r5d.2xlarge	emr-5.13.0, emr-6.0.0
	r5d.4xlarge	emr-5.13.0, emr-6.0.0
	r5d.8xlarge	emr-5.13.0, emr-6.0.0
	r5d.12xlarge	emr-5.13.0, emr-6.0.0
	r5d.16xlarge	emr-5.13.0, emr-6.0.0
	r5d.24xlarge	emr-5.13.0, emr-6.0.0
	r5dn.xlarge	emr-5.31.0, emr-6.1.0

Kelas Instans	Tipe instans	Versi EMR Amazon yang didukung minimum (5.x, 6.x, 7.x)
	r5dn.2xlarge	emr-5.31.0, emr-6.1.0
	r5dn.4xlarge	emr-5.31.0, emr-6.1.0
	r5dn.8xlarge	emr-5.31.0, emr-6.1.0
	r5dn.12xlarge	emr-5.31.0, emr-6.1.0
	r5dn.16xlarge	emr-5.31.0, emr-6.1.0
	r5dn.24xlarge	emr-5.31.0, emr-6.1.0
	r5n.xlarge	emr-5.31.0, emr-6.1.0
	r5n.2xlarge	emr-5.31.0, emr-6.1.0
	r5n.4xlarge	emr-5.31.0, emr-6.1.0
	r5n.8xlarge	emr-5.31.0, emr-6.1.0
	r5n.12xlarge	emr-5.31.0, emr-6.1.0
	r5n.16xlarge	emr-5.31.0, emr-6.1.0
	r5n.24xlarge	emr-5.31.0, emr-6.1.0
	r6a.xlarge	emr-5.36.1, emr-6.8.0
	r6a.2xlarge	emr-5.36.1, emr-6.8.0
	r6a.4xlarge	emr-5.36.1, emr-6.8.0
	r6a.8xlarge	emr-5.36.1, emr-6.8.0
	r6a.12xlarge	emr-5.36.1, emr-6.8.0
	r6a.16xlarge	emr-5.36.1, emr-6.8.0

Kelas Instans	Tipe instans	Versi EMR Amazon yang didukung minimum (5.x, 6.x, 7.x)
	r6a.24xlarge	emr-4.2.0, emr-5.0.0, emr-6.0.0
	r6a.32xlarge	emr-4.2.0, emr-5.0.0, emr-6.0.0
	r6a.48xlarge	emr-4.2.0, emr-5.0.0, emr-6.0.0
	r6g.xlarge	emr-5.31.0, emr-6.1.0
	r6g.2xlarge	emr-5.31.0, emr-6.1.0
	r6g.4xlarge	emr-5.31.0, emr-6.1.0
	r6g.8xlarge	emr-5.31.0, emr-6.1.0
	r6g.12xlarge	emr-5.31.0, emr-6.1.0
	r6g.16xlarge	emr-5.31.0, emr-6.1.0
	r6gd.xlarge	emr-5.33.0, emr-6.3.0
	r6gd.2xlarge	emr-5.33.0, emr-6.3.0
	r6gd.4xlarge	emr-5.33.0, emr-6.3.0
	r6gd.8xlarge	emr-5.33.0, emr-6.3.0
	r6gd.12xlarge	emr-5.33.0, emr-6.3.0
	r6gd.16xlarge	emr-5.33.0, emr-6.3.0
	r6i.xlarge	emr-5.35.0, emr-6.6.0
	r6i.2xlarge	emr-5.35.0, emr-6.6.0

Kelas Instans	Tipe instans	Versi EMR Amazon yang didukung minimum (5.x, 6.x, 7.x)
	r6i.4xlarge	emr-5.35.0, emr-6.6.0
	r6i.8xlarge	emr-5.35.0, emr-6.6.0
	r6i.12xlarge	emr-5.35.0, emr-6.6.0
	r6i.16xlarge	emr-5.35.0, emr-6.6.0
	r6i.24xlarge	emr-5.35.0, emr-6.6.0
	r6i.32xlarge	emr-5.35.0, emr-6.6.0
	r6id.xlarge	emr-5.36.1, emr-6.8.0
	r6id.2xlarge	emr-5.36.1, emr-6.8.0
	r6id.4xlarge	emr-5.36.1, emr-6.8.0
	r6id.8xlarge	emr-5.36.1, emr-6.8.0
	r6id.12xlarge	emr-5.36.1, emr-6.8.0
	r6id.16xlarge	emr-5.36.1, emr-6.8.0
	r6id.24xlarge	emr-5.36.1, emr-6.8.0
	r6id.32xlarge	emr-5.36.1, emr-6.8.0
	r7g.xlarge	emr-5.36.1, emr-6.10.0
	r7g.2xlarge	emr-5.36.1, emr-6.10.0
	r7g.4xlarge	emr-5.36.1, emr-6.10.0
	r7g.8xlarge	emr-5.36.1, emr-6.10.0
	r7g.12xlarge	emr-5.36.1, emr-6.10.0

Kelas Instans	Tipe instans	Versi EMR Amazon yang didukung minimum (5.x, 6.x, 7.x)
	r7g.16xlarge	emr-5.36.1, emr-6.10.0
	r7gd.xlarge	emr-5.36.1, emr-6.10.0
	r7gd.2xlarge	emr-5.36.1, emr-6.10.0
	r7gd.4xlarge	emr-5.36.1, emr-6.10.0
	r7gd.8xlarge	emr-5.36.1, emr-6.10.0
	r7gd.12xlarge	emr-5.36.1, emr-6.10.0
	r7gd.16xlarge	emr-5.36.1, emr-6.10.0
	x1.16xlarge	emr-5.36.1, emr-6.9.0
	x1.32xlarge	emr-5.36.1, emr-6.9.0
	x1e.xlarge	emr-5.36.1, emr-6.10.0
	x1e.2xlarge	emr-5.36.1, emr-6.10.0
	x1e.4xlarge	emr-5.36.1, emr-6.10.0
	x1e.8xlarge	emr-5.36.1, emr-6.10.0
	x1e.16xlarge	emr-5.36.1, emr-6.10.0
	x1e.32xlarge	emr-5.36.1, emr-6.10.0
	x2idn.16xlarge	emr-5.36.1, emr-6.9.0
	x2idn.24xlarge	emr-5.36.1, emr-6.9.0
	x2idn.32xlarge	emr-5.36.1, emr-6.9.0
	x2iedn.xlarge	emr-5.36.1, emr-6.9.0

Kelas Instans	Tipe instans	Versi EMR Amazon yang didukung minimum (5.x, 6.x, 7.x)
	x2iedn.2xlarge	emr-5.36.1, emr-6.9.0
	x2iedn.4xlarge	emr-5.36.1, emr-6.9.0
	x2iedn.8xlarge	emr-5.36.1, emr-6.9.0
	x2iedn.16xlarge	emr-5.36.1, emr-6.9.0
	x2iedn.24xlarge	emr-5.36.1, emr-6.9.0
	x2iedn.32xlarge	emr-5.36.1, emr-6.9.0
	z1d.xlarge	emr-5.17.0, emr-6.0.0
	z1d.2xlarge	emr-5.17.0, emr-6.0.0
	z1d.3xlarge	emr-5.17.0, emr-6.0.0
	z1d.6xlarge	emr-5.17.0, emr-6.0.0
	z1d.12xlarge	emr-5.17.0, emr-6.0.0
Penyimpanan Dioptimalkan	d3.xlarge	emr-5.33.0, emr-6.3.0
	d3.2xlarge	emr-5.33.0, emr-6.3.0
	d3.4xlarge	emr-5.33.0, emr-6.3.0
	d3.8xlarge	emr-5.33.0, emr-6.3.0
	i3.xlarge	emr-5.9.0, emr-6.0.0
	i3.2xlarge	emr-5.9.0, emr-6.0.0
	i3.4xlarge	emr-5.9.0, emr-6.0.0
	i3.8xlarge	emr-5.9.0, emr-6.0.0

Kelas Instans	Tipe instans	Versi EMR Amazon yang didukung minimum (5.x, 6.x, 7.x)
	i3.16xlarge	emr-5.9.0, emr-6.0.0
	i3en.xlarge	emr-5.25.0, emr-6.0.0
	i3en.2xlarge	emr-5.25.0, emr-6.0.0
	i3en.3xlarge	emr-5.25.0, emr-6.0.0
	i3en.6xlarge	emr-5.25.0, emr-6.0.0
	i3en.12xlarge	emr-5.25.0, emr-6.0.0
	i3en.24xlarge	emr-5.25.0, emr-6.0.0
	i4i.xlarge	emr-5.36.1, emr-6.8.0
	i4i.2xlarge	emr-5.36.1, emr-6.8.0
	i4i.4xlarge	emr-5.36.1, emr-6.8.0
	i4i.8xlarge	emr-5.36.1, emr-6.8.0
	i4i.12xlarge	emr-5.36.1, emr-6.10.0
	i4i.16xlarge	emr-5.36.1, emr-6.8.0
	i4i.24xlarge	emr-5.36.1, emr-6.10.0
	i4i.32xlarge	emr-5.36.1, emr-6.8.0
	im4gn.xlarge	emr-5.35.0, emr-6.6.0
	im4gn.2xlarge	emr-5.35.0, emr-6.6.0
	im4gn.4xlarge	emr-5.35.0, emr-6.6.0
	im4gn.8xlarge	emr-5.35.0, emr-6.6.0

Kelas Instans	Tipe instans	Versi EMR Amazon yang didukung minimum (5.x, 6.x, 7.x)
	im4gn.16xlarge	emr-5.35.0, emr-6.6.0
	is4gen.xlarge	emr-5.35.0, emr-6.6.0
	is4gen.2xlarge	emr-5.35.0, emr-6.6.0
	is4gen.4xlarge	emr-5.35.0, emr-6.6.0
	is4gen.8xlarge	emr-5.35.0, emr-6.6.0

Asia Pacific (Tokyo) - ap-northeast-1

Kelas Instans	Tipe instans	Versi EMR Amazon yang didukung minimum (5.x, 6.x, 7.x)
Tujuan Umum	m5.xlarge	emr-5.13.0, emr-6.0.0, emr-7.0.0
	m5.2xlarge	emr-5.13.0, emr-6.0.0, emr-7.0.0
	m5.4xlarge	emr-5.13.0, emr-6.0.0, emr-7.0.0
	m5.8xlarge	emr-5.13.0, emr-6.0.0, emr-7.0.0
	m5.12xlarge	emr-5.13.0, emr-6.0.0, emr-7.0.0
	m5.16xlarge	emr-5.13.0, emr-6.0.0, emr-7.0.0

Kelas Instans	Tipe instans	Versi EMR Amazon yang didukung minimum (5.x, 6.x, 7.x)
	m5.24xlarge	emr-5.13.0, emr-6.0.0, emr-7.0.0
	m5a.xlarge	emr-5.20.0, emr-6.0.0, emr-7.0.0
	m5a.2xlarge	emr-5.20.0, emr-6.0.0, emr-7.0.0
	m5a.4xlarge	emr-5.20.0, emr-6.0.0, emr-7.0.0
	m5a.8xlarge	emr-5.20.0, emr-6.0.0, emr-7.0.0
	m5a.12xlarge	emr-5.20.0, emr-6.0.0, emr-7.0.0
	m5a.16xlarge	emr-5.20.0, emr-6.0.0, emr-7.0.0
	m5a.24xlarge	emr-5.20.0, emr-6.0.0, emr-7.0.0
	m5ad.xlarge	emr-5.20.0, emr-6.0.0, emr-7.0.0
	m5ad.2xlarge	emr-5.20.0, emr-6.0.0, emr-7.0.0
	m5ad.4xlarge	emr-5.20.0, emr-6.0.0, emr-7.0.0
	m5ad.8xlarge	emr-5.33.0, emr-6.3.0, emr-7.0.0

Kelas Instans	Tipe instans	Versi EMR Amazon yang didukung minimum (5.x, 6.x, 7.x)
	m5ad.12xlarge	emr-5.20.0, emr-6.0.0, emr-7.0.0
	m5ad.16xlarge	emr-5.33.0, emr-6.3.0, emr-7.0.0
	m5ad.24xlarge	emr-5.20.0, emr-6.0.0, emr-7.0.0
	m5d.xlarge	emr-5.13.0, emr-6.0.0, emr-7.0.0
	m5d.2xlarge	emr-5.13.0, emr-6.0.0, emr-7.0.0
	m5d.4xlarge	emr-5.13.0, emr-6.0.0, emr-7.0.0
	m5d.8xlarge	emr-5.13.0, emr-6.0.0, emr-7.0.0
	m5d.12xlarge	emr-5.13.0, emr-6.0.0, emr-7.0.0
	m5d.16xlarge	emr-5.13.0, emr-6.0.0, emr-7.0.0
	m5d.24xlarge	emr-5.13.0, emr-6.0.0, emr-7.0.0
	m5dn.xlarge	emr-5.31.0, emr-6.1.0, emr-7.0.0
	m5dn.2xlarge	emr-5.31.0, emr-6.1.0, emr-7.0.0

Kelas Instans	Tipe instans	Versi EMR Amazon yang didukung minimum (5.x, 6.x, 7.x)
	m5dn.4xlarge	emr-5.31.0, emr-6.1.0, emr-7.0.0
	m5dn.8xlarge	emr-5.31.0, emr-6.1.0, emr-7.0.0
	m5dn.12xlarge	emr-5.31.0, emr-6.1.0, emr-7.0.0
	m5dn.16xlarge	emr-5.31.0, emr-6.1.0, emr-7.0.0
	m5dn.24xlarge	emr-5.31.0, emr-6.1.0, emr-7.0.0
	m5n.xlarge	emr-5.31.0, emr-6.1.0, emr-7.0.0
	m5n.2xlarge	emr-5.31.0, emr-6.1.0, emr-7.0.0
	m5n.4xlarge	emr-5.31.0, emr-6.1.0, emr-7.0.0
	m5n.8xlarge	emr-5.31.0, emr-6.1.0, emr-7.0.0
	m5n.12xlarge	emr-5.31.0, emr-6.1.0, emr-7.0.0
	m5n.16xlarge	emr-5.31.0, emr-6.1.0, emr-7.0.0
	m5n.24xlarge	emr-5.31.0, emr-6.1.0, emr-7.0.0

Kelas Instans	Tipe instans	Versi EMR Amazon yang didukung minimum (5.x, 6.x, 7.x)
	m5zn.xlarge	emr-5.33.0, emr-6.3.0, emr-7.0.0
	m5zn.2xlarge	emr-5.33.0, emr-6.3.0, emr-7.0.0
	m5zn.3xlarge	emr-5.33.0, emr-6.3.0, emr-7.0.0
	m5zn.6xlarge	emr-5.33.0, emr-6.3.0, emr-7.0.0
	m5zn.12xlarge	emr-5.33.0, emr-6.3.0, emr-7.0.0
	m6a.xlarge	emr-5.36.1, emr-6.8.0, emr-7.0.0
	m6a.2xlarge	emr-5.36.1, emr-6.8.0, emr-7.0.0
	m6a.4xlarge	emr-5.36.1, emr-6.8.0, emr-7.0.0
	m6a.8xlarge	emr-5.36.1, emr-6.8.0, emr-7.0.0
	m6a.12xlarge	emr-5.36.1, emr-6.8.0, emr-7.0.0
	m6a.16xlarge	emr-5.36.1, emr-6.8.0, emr-7.0.0
	m6a.24xlarge	emr-5.36.1, emr-6.8.0, emr-7.0.0

Kelas Instans	Tipe instans	Versi EMR Amazon yang didukung minimum (5.x, 6.x, 7.x)
	m6a.32xlarge	emr-5.36.1, emr-6.8.0, emr-7.0.0
	m6a.48xlarge	emr-5.36.1, emr-6.8.0, emr-7.0.0
	m6g.xlarge	emr-5.30.0, emr-6.1.0, emr-7.0.0
	m6g.2xlarge	emr-5.30.0, emr-6.1.0, emr-7.0.0
	m6g.4xlarge	emr-5.30.0, emr-6.1.0, emr-7.0.0
	m6g.8xlarge	emr-5.30.0, emr-6.1.0, emr-7.0.0
	m6g.12xlarge	emr-5.30.0, emr-6.1.0, emr-7.0.0
	m6g.16xlarge	emr-5.30.0, emr-6.1.0, emr-7.0.0
	m6gd.xlarge	emr-5.33.0, emr-6.3.0, emr-7.0.0
	m6gd.2xlarge	emr-5.33.0, emr-6.3.0, emr-7.0.0
	m6gd.4xlarge	emr-5.33.0, emr-6.3.0, emr-7.0.0
	m6gd.8xlarge	emr-5.33.0, emr-6.3.0, emr-7.0.0

Kelas Instans	Tipe instans	Versi EMR Amazon yang didukung minimum (5.x, 6.x, 7.x)
	m6gd.12xlarge	emr-5.33.0, emr-6.3.0, emr-7.0.0
	m6gd.16xlarge	emr-5.33.0, emr-6.3.0, emr-7.0.0
	m6i.xlarge	emr-5.35.0, emr-6.6.0
	m6i.2xlarge	emr-5.35.0, emr-6.6.0
	m6i.4xlarge	emr-5.35.0, emr-6.6.0
	m6i.8xlarge	emr-5.35.0, emr-6.6.0
	m6i.12xlarge	emr-5.35.0, emr-6.6.0
	m6i.16xlarge	emr-5.35.0, emr-6.6.0
	m6i.24xlarge	emr-5.35.0, emr-6.6.0
	m6i.32xlarge	emr-5.35.0, emr-6.6.0
	m6id.xlarge	emr-5.36.1, emr-6.8.0
	m6id.2xlarge	emr-5.36.1, emr-6.8.0
	m6id.4xlarge	emr-5.36.1, emr-6.8.0
	m6id.8xlarge	emr-5.36.1, emr-6.8.0
	m6id.12xlarge	emr-5.36.1, emr-6.8.0
	m6id.16xlarge	emr-5.36.1, emr-6.8.0
	m6id.24xlarge	emr-5.36.1, emr-6.8.0
	m6id.32xlarge	emr-5.36.1, emr-6.8.0

Kelas Instans	Tipe instans	Versi EMR Amazon yang didukung minimum (5.x, 6.x, 7.x)
	m6idn.xlarge	emr-5.36.1, emr-6.10.0
	m6idn.2xlarge	emr-5.36.1, emr-6.10.0
	m6idn.4xlarge	emr-5.36.1, emr-6.10.0
	m6idn.8xlarge	emr-5.36.1, emr-6.10.0
	m6idn.12xlarge	emr-5.36.1, emr-6.10.0
	m6idn.16xlarge	emr-5.36.1, emr-6.10.0
	m6idn.24xlarge	emr-5.36.1, emr-6.10.0
	m6idn.32xlarge	emr-5.36.1, emr-6.10.0
	m6in.xlarge	emr-5.36.1, emr-6.10.0
	m6in.2xlarge	emr-5.36.1, emr-6.10.0
	m6in.4xlarge	emr-5.36.1, emr-6.10.0
	m6in.8xlarge	emr-5.36.1, emr-6.10.0
	m6in.12xlarge	emr-5.36.1, emr-6.10.0
	m6in.16xlarge	emr-5.36.1, emr-6.10.0
	m6in.24xlarge	emr-5.36.1, emr-6.10.0
	m6in.32xlarge	emr-5.36.1, emr-6.10.0
	m7g.xlarge	emr-5.36.1, emr-6.10.0, emr-7.0.0
	m7g.2xlarge	emr-5.36.1, emr-6.10.0, emr-7.0.0

Kelas Instans	Tipe instans	Versi EMR Amazon yang didukung minimum (5.x, 6.x, 7.x)
	m7g.4xlarge	emr-5.36.1, emr-6.10.0, emr-7.0.0
	m7g.8xlarge	emr-5.36.1, emr-6.10.0, emr-7.0.0
	m7g.12xlarge	emr-5.36.1, emr-6.10.0, emr-7.0.0
	m7g.16xlarge	emr-5.36.1, emr-6.10.0, emr-7.0.0
	m7gd.xlarge	emr-5.36.1, emr-6.10.0, emr-7.0.0
	m7gd.2xlarge	emr-5.36.1, emr-6.10.0, emr-7.0.0
	m7gd.4xlarge	emr-5.36.1, emr-6.10.0, emr-7.0.0
	m7gd.8xlarge	emr-5.36.1, emr-6.10.0, emr-7.0.0
	m7gd.12xlarge	emr-5.36.1, emr-6.10.0, emr-7.0.0
	m7gd.16xlarge	emr-5.36.1, emr-6.10.0, emr-7.0.0
Komputasi Dioptimalkan	c5.xlarge	emr-5.13.0, emr-6.0.0, emr-7.0.0
	c5.2xlarge	emr-5.13.0, emr-6.0.0, emr-7.0.0

Kelas Instans	Tipe instans	Versi EMR Amazon yang didukung minimum (5.x, 6.x, 7.x)
	c5.4xlarge	emr-5.13.0, emr-6.0.0, emr-7.0.0
	c5.9xlarge	emr-5.13.0, emr-6.0.0, emr-7.0.0
	c5.12xlarge	emr-5.13.0, emr-6.0.0, emr-7.0.0
	c5.18xlarge	emr-5.13.0, emr-6.0.0, emr-7.0.0
	c5.24xlarge	emr-5.13.0, emr-6.0.0, emr-7.0.0
	c5a.xlarge	emr-5.31.0, emr-6.1.0, emr-7.0.0
	c5a.2xlarge	emr-5.31.0, emr-6.1.0, emr-7.0.0
	c5a.4xlarge	emr-5.31.0, emr-6.1.0, emr-7.0.0
	c5a.8xlarge	emr-5.31.0, emr-6.1.0, emr-7.0.0
	c5a.12xlarge	emr-5.31.0, emr-6.1.0, emr-7.0.0
	c5a.16xlarge	emr-5.31.0, emr-6.1.0, emr-7.0.0
	c5a.24xlarge	emr-5.31.0, emr-6.1.0, emr-7.0.0

Kelas Instans	Tipe instans	Versi EMR Amazon yang didukung minimum (5.x, 6.x, 7.x)
	c5d.xlarge	emr-5.13.0, emr-6.0.0, emr-7.0.0
	c5d.2xlarge	emr-5.13.0, emr-6.0.0, emr-7.0.0
	c5d.4xlarge	emr-5.13.0, emr-6.0.0, emr-7.0.0
	c5d.9xlarge	emr-5.13.0, emr-6.0.0, emr-7.0.0
	c5d.12xlarge	emr-5.13.0, emr-6.0.0, emr-7.0.0
	c5d.18xlarge	emr-5.13.0, emr-6.0.0, emr-7.0.0
	c5d.24xlarge	emr-5.13.0, emr-6.0.0, emr-7.0.0
	c5n.xlarge	emr-5.20.0, emr-6.0.0, emr-7.0.0
	c5n.2xlarge	emr-5.20.0, emr-6.0.0, emr-7.0.0
	c5n.4xlarge	emr-5.20.0, emr-6.0.0, emr-7.0.0
	c5n.9xlarge	emr-5.20.0, emr-6.0.0, emr-7.0.0
	c5n.18xlarge	emr-5.20.0, emr-6.0.0, emr-7.0.0

Kelas Instans	Tipe instans	Versi EMR Amazon yang didukung minimum (5.x, 6.x, 7.x)
	c6a.xlarge	emr-5.36.1, emr-6.8.0, emr-7.0.0
	c6a.2xlarge	emr-5.36.1, emr-6.8.0, emr-7.0.0
	c6a.4xlarge	emr-5.36.1, emr-6.8.0, emr-7.0.0
	c6a.8xlarge	emr-5.36.1, emr-6.8.0, emr-7.0.0
	c6a.12xlarge	emr-5.36.1, emr-6.8.0, emr-7.0.0
	c6a.16xlarge	emr-5.36.1, emr-6.8.0, emr-7.0.0
	c6a.24xlarge	emr-5.36.1, emr-6.8.0, emr-7.0.0
	c6a.32xlarge	emr-5.36.1, emr-6.8.0, emr-7.0.0
	c6a.48xlarge	emr-5.36.1, emr-6.8.0, emr-7.0.0
	c6g.xlarge	emr-5.31.0, emr-6.1.0, emr-7.0.0
	c6g.2xlarge	emr-5.31.0, emr-6.1.0, emr-7.0.0
	c6g.4xlarge	emr-5.31.0, emr-6.1.0, emr-7.0.0

Kelas Instans	Tipe instans	Versi EMR Amazon yang didukung minimum (5.x, 6.x, 7.x)
	c6g.8xlarge	emr-5.31.0, emr-6.1.0, emr-7.0.0
	c6g.12xlarge	emr-5.31.0, emr-6.1.0, emr-7.0.0
	c6g.16xlarge	emr-5.31.0, emr-6.1.0, emr-7.0.0
	c6gd.xlarge	emr-5.33.0, emr-6.3.0, emr-7.0.0
	c6gd.2xlarge	emr-5.33.0, emr-6.3.0, emr-7.0.0
	c6gd.4xlarge	emr-5.33.0, emr-6.3.0, emr-7.0.0
	c6gd.8xlarge	emr-5.33.0, emr-6.3.0, emr-7.0.0
	c6gd.12xlarge	emr-5.33.0, emr-6.3.0, emr-7.0.0
	c6gd.16xlarge	emr-5.33.0, emr-6.3.0, emr-7.0.0
	c6gn.xlarge	emr-5.33.0, emr-6.3.0, emr-7.0.0
	c6gn.2xlarge	emr-5.33.0, emr-6.3.0, emr-7.0.0
	c6gn.4xlarge	emr-5.33.0, emr-6.3.0, emr-7.0.0

Kelas Instans	Tipe instans	Versi EMR Amazon yang didukung minimum (5.x, 6.x, 7.x)
	c6gn.8xlarge	emr-5.33.0, emr-6.3.0, emr-7.0.0
	c6gn.12xlarge	emr-5.33.0, emr-6.3.0, emr-7.0.0
	c6gn.16xlarge	emr-5.33.0, emr-6.3.0, emr-7.0.0
	c6i.xlarge	emr-5.35.0, emr-6.6.0
	c6i.2xlarge	emr-5.35.0, emr-6.6.0
	c6i.4xlarge	emr-5.35.0, emr-6.6.0
	c6i.8xlarge	emr-5.35.0, emr-6.6.0
	c6i.12xlarge	emr-5.35.0, emr-6.6.0
	c6i.16xlarge	emr-5.35.0, emr-6.6.0
	c6i.24xlarge	emr-5.35.0, emr-6.6.0
	c6i.32xlarge	emr-5.35.0, emr-6.6.0
	c6id.xlarge	emr-5.36.1, emr-6.8.0
	c6id.2xlarge	emr-5.36.1, emr-6.8.0
	c6id.4xlarge	emr-5.36.1, emr-6.8.0
	c6id.8xlarge	emr-5.36.1, emr-6.8.0
	c6id.12xlarge	emr-5.36.1, emr-6.8.0
	c6id.16xlarge	emr-5.36.1, emr-6.8.0

Kelas Instans	Tipe instans	Versi EMR Amazon yang didukung minimum (5.x, 6.x, 7.x)
	c6id.24xlarge	emr-5.36.1, emr-6.8.0
	c6id.32xlarge	emr-5.36.1, emr-6.8.0
	c6in.xlarge	emr-5.36.1, emr-6.10.0
	c6in.2xlarge	emr-5.36.1, emr-6.10.0
	c6in.4xlarge	emr-5.36.1, emr-6.10.0
	c6in.8xlarge	emr-5.36.1, emr-6.10.0
	c6in.12xlarge	emr-5.36.1, emr-6.10.0
	c6in.16xlarge	emr-5.36.1, emr-6.10.0
	c6in.24xlarge	emr-5.36.1, emr-6.10.0
	c6in.32xlarge	emr-5.36.1, emr-6.10.0
	c7g.xlarge	emr-5.36.1, emr-6.7.0, emr-7.0.0
	c7g.2xlarge	emr-5.36.1, emr-6.7.0, emr-7.0.0
	c7g.4xlarge	emr-5.36.1, emr-6.7.0, emr-7.0.0
	c7g.8xlarge	emr-5.36.1, emr-6.7.0, emr-7.0.0
	c7g.12xlarge	emr-5.36.1, emr-6.7.0, emr-7.0.0
	c7g.16xlarge	emr-5.36.1, emr-6.7.0, emr-7.0.0

Kelas Instans	Tipe instans	Versi EMR Amazon yang didukung minimum (5.x, 6.x, 7.x)
	c7gd.xlarge	emr-5.36.1, emr-6.10.0, emr-7.0.0
	c7gd.2xlarge	emr-5.36.1, emr-6.10.0, emr-7.0.0
	c7gd.4xlarge	emr-5.36.1, emr-6.10.0, emr-7.0.0
	c7gd.8xlarge	emr-5.36.1, emr-6.10.0, emr-7.0.0
	c7gd.12xlarge	emr-5.36.1, emr-6.10.0, emr-7.0.0
	c7gd.16xlarge	emr-5.36.1, emr-6.10.0, emr-7.0.0
	c7gn.xlarge	emr-5.36.1, emr-6.10.0, emr-7.0.0
	c7gn.2xlarge	emr-5.36.1, emr-6.10.0, emr-7.0.0
	c7gn.4xlarge	emr-5.36.1, emr-6.10.0, emr-7.0.0
	c7gn.8xlarge	emr-5.36.1, emr-6.10.0, emr-7.0.0
	c7gn.12xlarge	emr-5.36.1, emr-6.10.0, emr-7.0.0
	c7gn.16xlarge	emr-5.36.1, emr-6.10.0, emr-7.0.0

Kelas Instans	Tipe instans	Versi EMR Amazon yang didukung minimum (5.x, 6.x, 7.x)
Komputasi yang Dipercepat	g3.4xlarge	emr-5.18.0, emr-6.0.0, emr-7.0.0
	g3.8xlarge	emr-5.18.0, emr-6.0.0, emr-7.0.0
	g3.16xlarge	emr-5.18.0, emr-6.0.0, emr-7.0.0
	g3s.xlarge	emr-5.19.0, emr-6.0.0, emr-7.0.0
	g4dn.xlarge	emr-5.30.0, emr-6.0.0, emr-7.0.0
	g4dn.2xlarge	emr-5.30.0, emr-6.0.0, emr-7.0.0
	g4dn.4xlarge	emr-5.30.0, emr-6.0.0, emr-7.0.0
	g4dn.8xlarge	emr-5.30.0, emr-6.0.0, emr-7.0.0
	g4dn.12xlarge	emr-5.30.0, emr-6.0.0, emr-7.0.0
	g4dn.16xlarge	emr-5.30.0, emr-6.0.0, emr-7.0.0
	g5.xlarge	emr-5.36.1, emr-6.9.0, emr-7.0.0
	g5.2xlarge	emr-5.36.1, emr-6.9.0, emr-7.0.0

Kelas Instans	Tipe instans	Versi EMR Amazon yang didukung minimum (5.x, 6.x, 7.x)
	g5.4xlarge	emr-5.36.1, emr-6.9.0, emr-7.0.0
	g5.8xlarge	emr-5.36.1, emr-6.9.0, emr-7.0.0
	g5.12xlarge	emr-5.36.1, emr-6.9.0, emr-7.0.0
	g5.16xlarge	emr-5.36.1, emr-6.9.0, emr-7.0.0
	g5.24xlarge	emr-5.36.1, emr-6.9.0, emr-7.0.0
	g5.48xlarge	emr-5.36.1, emr-6.9.0, emr-7.0.0
	p2.xlarge	emr-5.10.0, emr-6.0.0, emr-7.0.0
	p2.8xlarge	emr-5.10.0, emr-6.0.0, emr-7.0.0
	p2.16xlarge	emr-5.10.0, emr-6.0.0, emr-7.0.0
	p3.2xlarge	emr-5.10.0, emr-6.0.0, emr-7.0.0
	p3.8xlarge	emr-5.10.0, emr-6.0.0, emr-7.0.0
	p3.16xlarge	emr-5.10.0, emr-6.0.0, emr-7.0.0

Kelas Instans	Tipe instans	Versi EMR Amazon yang didukung minimum (5.x, 6.x, 7.x)
Memori Dioptimalkan	r5.xlarge	emr-5.13.0, emr-6.0.0, emr-7.0.0
	r5.2xlarge	emr-5.13.0, emr-6.0.0, emr-7.0.0
	r5.4xlarge	emr-5.13.0, emr-6.0.0, emr-7.0.0
	r5.8xlarge	emr-5.13.0, emr-6.0.0, emr-7.0.0
	r5.12xlarge	emr-5.13.0, emr-6.0.0, emr-7.0.0
	r5.16xlarge	emr-5.13.0, emr-6.0.0, emr-7.0.0
	r5.24xlarge	emr-5.13.0, emr-6.0.0, emr-7.0.0
	r5a.xlarge	emr-5.20.0, emr-6.0.0, emr-7.0.0
	r5a.2xlarge	emr-5.20.0, emr-6.0.0, emr-7.0.0
	r5a.4xlarge	emr-5.20.0, emr-6.0.0, emr-7.0.0
	r5a.8xlarge	emr-5.20.0, emr-6.0.0, emr-7.0.0
	r5a.12xlarge	emr-5.20.0, emr-6.0.0, emr-7.0.0

Kelas Instans	Tipe instans	Versi EMR Amazon yang didukung minimum (5.x, 6.x, 7.x)
	r5a.16xlarge	emr-5.20.0, emr-6.0.0, emr-7.0.0
	r5a.24xlarge	emr-5.20.0, emr-6.0.0, emr-7.0.0
	r5ad.xlarge	emr-5.20.0, emr-6.0.0, emr-7.0.0
	r5ad.2xlarge	emr-5.20.0, emr-6.0.0, emr-7.0.0
	r5ad.4xlarge	emr-5.20.0, emr-6.0.0, emr-7.0.0
	r5ad.8xlarge	emr-5.33.0, emr-6.3.0, emr-7.0.0
	r5ad.12xlarge	emr-5.20.0, emr-6.0.0, emr-7.0.0
	r5ad.16xlarge	emr-5.33.0, emr-6.3.0, emr-7.0.0
	r5ad.24xlarge	emr-5.20.0, emr-6.0.0, emr-7.0.0
	r5b.xlarge	emr-5.33.0, emr-6.3.0, emr-7.0.0
	r5b.2xlarge	emr-5.33.0, emr-6.3.0, emr-7.0.0
	r5b.4xlarge	emr-5.33.0, emr-6.3.0, emr-7.0.0

Kelas Instans	Tipe instans	Versi EMR Amazon yang didukung minimum (5.x, 6.x, 7.x)
	r5b.8xlarge	emr-5.33.0, emr-6.3.0, emr-7.0.0
	r5b.12xlarge	emr-5.33.0, emr-6.3.0, emr-7.0.0
	r5b.16xlarge	emr-5.33.0, emr-6.3.0, emr-7.0.0
	r5b.24xlarge	emr-5.33.0, emr-6.3.0, emr-7.0.0
	r5d.xlarge	emr-5.13.0, emr-6.0.0, emr-7.0.0
	r5d.2xlarge	emr-5.13.0, emr-6.0.0, emr-7.0.0
	r5d.4xlarge	emr-5.13.0, emr-6.0.0, emr-7.0.0
	r5d.8xlarge	emr-5.13.0, emr-6.0.0, emr-7.0.0
	r5d.12xlarge	emr-5.13.0, emr-6.0.0, emr-7.0.0
	r5d.16xlarge	emr-5.13.0, emr-6.0.0, emr-7.0.0
	r5d.24xlarge	emr-5.13.0, emr-6.0.0, emr-7.0.0
	r5dn.xlarge	emr-5.31.0, emr-6.1.0, emr-7.0.0

Kelas Instans	Tipe instans	Versi EMR Amazon yang didukung minimum (5.x, 6.x, 7.x)
	r5dn.2xlarge	emr-5.31.0, emr-6.1.0, emr-7.0.0
	r5dn.4xlarge	emr-5.31.0, emr-6.1.0, emr-7.0.0
	r5dn.8xlarge	emr-5.31.0, emr-6.1.0, emr-7.0.0
	r5dn.12xlarge	emr-5.31.0, emr-6.1.0, emr-7.0.0
	r5dn.16xlarge	emr-5.31.0, emr-6.1.0, emr-7.0.0
	r5dn.24xlarge	emr-5.31.0, emr-6.1.0, emr-7.0.0
	r5n.xlarge	emr-5.31.0, emr-6.1.0, emr-7.0.0
	r5n.2xlarge	emr-5.31.0, emr-6.1.0, emr-7.0.0
	r5n.4xlarge	emr-5.31.0, emr-6.1.0, emr-7.0.0
	r5n.8xlarge	emr-5.31.0, emr-6.1.0, emr-7.0.0
	r5n.12xlarge	emr-5.31.0, emr-6.1.0, emr-7.0.0
	r5n.16xlarge	emr-5.31.0, emr-6.1.0, emr-7.0.0

Kelas Instans	Tipe instans	Versi EMR Amazon yang didukung minimum (5.x, 6.x, 7.x)
	r5n.24xlarge	emr-5.31.0, emr-6.1.0, emr-7.0.0
	r6a.xlarge	emr-5.36.1, emr-6.8.0, emr-7.0.0
	r6a.2xlarge	emr-5.36.1, emr-6.8.0, emr-7.0.0
	r6a.4xlarge	emr-5.36.1, emr-6.8.0, emr-7.0.0
	r6a.8xlarge	emr-5.36.1, emr-6.8.0, emr-7.0.0
	r6a.12xlarge	emr-5.36.1, emr-6.8.0, emr-7.0.0
	r6a.16xlarge	emr-5.36.1, emr-6.8.0, emr-7.0.0
	r6a.24xlarge	emr-4.2.0, emr-5.0.0, emr-6.0.0, emr-7.0.0
	r6a.32xlarge	emr-4.2.0, emr-5.0.0, emr-6.0.0, emr-7.0.0
	r6a.48xlarge	emr-4.2.0, emr-5.0.0, emr-6.0.0, emr-7.0.0
	r6g.xlarge	emr-5.31.0, emr-6.1.0, emr-7.0.0
	r6g.2xlarge	emr-5.31.0, emr-6.1.0, emr-7.0.0

Kelas Instans	Tipe instans	Versi EMR Amazon yang didukung minimum (5.x, 6.x, 7.x)
	r6g.4xlarge	emr-5.31.0, emr-6.1.0, emr-7.0.0
	r6g.8xlarge	emr-5.31.0, emr-6.1.0, emr-7.0.0
	r6g.12xlarge	emr-5.31.0, emr-6.1.0, emr-7.0.0
	r6g.16xlarge	emr-5.31.0, emr-6.1.0, emr-7.0.0
	r6gd.xlarge	emr-5.33.0, emr-6.3.0, emr-7.0.0
	r6gd.2xlarge	emr-5.33.0, emr-6.3.0, emr-7.0.0
	r6gd.4xlarge	emr-5.33.0, emr-6.3.0, emr-7.0.0
	r6gd.8xlarge	emr-5.33.0, emr-6.3.0, emr-7.0.0
	r6gd.12xlarge	emr-5.33.0, emr-6.3.0, emr-7.0.0
	r6gd.16xlarge	emr-5.33.0, emr-6.3.0, emr-7.0.0
	r6i.xlarge	emr-5.35.0, emr-6.6.0
	r6i.2xlarge	emr-5.35.0, emr-6.6.0
	r6i.4xlarge	emr-5.35.0, emr-6.6.0

Kelas Instans	Tipe instans	Versi EMR Amazon yang didukung minimum (5.x, 6.x, 7.x)
	r6i.8xlarge	emr-5.35.0, emr-6.6.0
	r6i.12xlarge	emr-5.35.0, emr-6.6.0
	r6i.16xlarge	emr-5.35.0, emr-6.6.0
	r6i.24xlarge	emr-5.35.0, emr-6.6.0
	r6i.32xlarge	emr-5.35.0, emr-6.6.0
	r6id.xlarge	emr-5.36.1, emr-6.8.0
	r6id.2xlarge	emr-5.36.1, emr-6.8.0
	r6id.4xlarge	emr-5.36.1, emr-6.8.0
	r6id.8xlarge	emr-5.36.1, emr-6.8.0
	r6id.12xlarge	emr-5.36.1, emr-6.8.0
	r6id.16xlarge	emr-5.36.1, emr-6.8.0
	r6id.24xlarge	emr-5.36.1, emr-6.8.0
	r6id.32xlarge	emr-5.36.1, emr-6.8.0
	r6idn.xlarge	emr-5.36.1, emr-6.10.0
	r6idn.2xlarge	emr-5.36.1, emr-6.10.0
	r6idn.4xlarge	emr-5.36.1, emr-6.10.0
	r6idn.8xlarge	emr-5.36.1, emr-6.10.0
	r6idn.12xlarge	emr-5.36.1, emr-6.10.0
	r6idn.16xlarge	emr-5.36.1, emr-6.10.0

Kelas Instans	Tipe instans	Versi EMR Amazon yang didukung minimum (5.x, 6.x, 7.x)
	r6idn.24xlarge	emr-5.36.1, emr-6.10.0
	r6idn.32xlarge	emr-5.36.1, emr-6.10.0
	r6in.xlarge	emr-5.36.1, emr-6.10.0
	r6in.2xlarge	emr-5.36.1, emr-6.10.0
	r6in.4xlarge	emr-5.36.1, emr-6.10.0
	r6in.8xlarge	emr-5.36.1, emr-6.10.0
	r6in.12xlarge	emr-5.36.1, emr-6.10.0
	r6in.16xlarge	emr-5.36.1, emr-6.10.0
	r6in.24xlarge	emr-5.36.1, emr-6.10.0
	r6in.32xlarge	emr-5.36.1, emr-6.10.0
	r7g.xlarge	emr-5.36.1, emr-6.10.0, emr-7.0.0
	r7g.2xlarge	emr-5.36.1, emr-6.10.0, emr-7.0.0
	r7g.4xlarge	emr-5.36.1, emr-6.10.0, emr-7.0.0
	r7g.8xlarge	emr-5.36.1, emr-6.10.0, emr-7.0.0
	r7g.12xlarge	emr-5.36.1, emr-6.10.0, emr-7.0.0
	r7g.16xlarge	emr-5.36.1, emr-6.10.0, emr-7.0.0

Kelas Instans	Tipe instans	Versi EMR Amazon yang didukung minimum (5.x, 6.x, 7.x)
	r7gd.xlarge	emr-5.36.1, emr-6.10.0, emr-7.0.0
	r7gd.2xlarge	emr-5.36.1, emr-6.10.0, emr-7.0.0
	r7gd.4xlarge	emr-5.36.1, emr-6.10.0, emr-7.0.0
	r7gd.8xlarge	emr-5.36.1, emr-6.10.0, emr-7.0.0
	r7gd.12xlarge	emr-5.36.1, emr-6.10.0, emr-7.0.0
	r7gd.16xlarge	emr-5.36.1, emr-6.10.0, emr-7.0.0
	x1.16xlarge	emr-5.36.1, emr-6.9.0, emr-7.0.0
	x1.32xlarge	emr-5.36.1, emr-6.9.0, emr-7.0.0
	x1e.xlarge	emr-5.36.1, emr-6.10.0, emr-7.0.0
	x1e.2xlarge	emr-5.36.1, emr-6.10.0, emr-7.0.0
	x1e.4xlarge	emr-5.36.1, emr-6.10.0, emr-7.0.0
	x1e.8xlarge	emr-5.36.1, emr-6.10.0, emr-7.0.0

Kelas Instans	Tipe instans	Versi EMR Amazon yang didukung minimum (5.x, 6.x, 7.x)
	x1e.16xlarge	emr-5.36.1, emr-6.10.0, emr-7.0.0
	x1e.32xlarge	emr-5.36.1, emr-6.10.0, emr-7.0.0
	x2idn.16xlarge	emr-5.36.1, emr-6.9.0
	x2idn.24xlarge	emr-5.36.1, emr-6.9.0
	x2idn.32xlarge	emr-5.36.1, emr-6.9.0
	x2iedn.xlarge	emr-5.36.1, emr-6.9.0
	x2iedn.2xlarge	emr-5.36.1, emr-6.9.0
	x2iedn.4xlarge	emr-5.36.1, emr-6.9.0
	x2iedn.8xlarge	emr-5.36.1, emr-6.9.0
	x2iedn.16xlarge	emr-5.36.1, emr-6.9.0
	x2iedn.24xlarge	emr-5.36.1, emr-6.9.0
	x2iedn.32xlarge	emr-5.36.1, emr-6.9.0
	z1d.xlarge	emr-5.17.0, emr-6.0.0, emr-7.0.0
	z1d.2xlarge	emr-5.17.0, emr-6.0.0, emr-7.0.0
	z1d.3xlarge	emr-5.17.0, emr-6.0.0, emr-7.0.0
	z1d.6xlarge	emr-5.17.0, emr-6.0.0, emr-7.0.0

Kelas Instans	Tipe instans	Versi EMR Amazon yang didukung minimum (5.x, 6.x, 7.x)
	z1d.12xlarge	emr-5.17.0, emr-6.0.0, emr-7.0.0
Penyimpanan Dioptimalkan	d3.xlarge	emr-5.33.0, emr-6.3.0, emr-7.0.0
	d3.2xlarge	emr-5.33.0, emr-6.3.0, emr-7.0.0
	d3.4xlarge	emr-5.33.0, emr-6.3.0, emr-7.0.0
	d3.8xlarge	emr-5.33.0, emr-6.3.0, emr-7.0.0
	d3en.xlarge	emr-5.33.0, emr-6.3.0, emr-7.0.0
	d3en.2xlarge	emr-5.33.0, emr-6.3.0, emr-7.0.0
	d3en.4xlarge	emr-5.33.0, emr-6.3.0, emr-7.0.0
	d3en.6xlarge	emr-5.33.0, emr-6.3.0, emr-7.0.0
	d3en.8xlarge	emr-5.33.0, emr-6.3.0, emr-7.0.0
	d3en.12xlarge	emr-5.33.0, emr-6.3.0, emr-7.0.0
	i3.xlarge	emr-5.9.0, emr-6.0.0, emr-7.0.0

Kelas Instans	Tipe instans	Versi EMR Amazon yang didukung minimum (5.x, 6.x, 7.x)
	i3.2xlarge	emr-5.9.0, emr-6.0.0, emr-7.0.0
	i3.4xlarge	emr-5.9.0, emr-6.0.0, emr-7.0.0
	i3.8xlarge	emr-5.9.0, emr-6.0.0, emr-7.0.0
	i3.16xlarge	emr-5.9.0, emr-6.0.0, emr-7.0.0
	i3en.xlarge	emr-5.25.0, emr-6.0.0, emr-7.0.0
	i3en.2xlarge	emr-5.25.0, emr-6.0.0, emr-7.0.0
	i3en.3xlarge	emr-5.25.0, emr-6.0.0, emr-7.0.0
	i3en.6xlarge	emr-5.25.0, emr-6.0.0, emr-7.0.0
	i3en.12xlarge	emr-5.25.0, emr-6.0.0, emr-7.0.0
	i3en.24xlarge	emr-5.25.0, emr-6.0.0, emr-7.0.0
	i4i.xlarge	emr-5.36.1, emr-6.8.0
	i4i.2xlarge	emr-5.36.1, emr-6.8.0
	i4i.4xlarge	emr-5.36.1, emr-6.8.0

Kelas Instans	Tipe instans	Versi EMR Amazon yang didukung minimum (5.x, 6.x, 7.x)
	i4i.8xlarge	emr-5.36.1, emr-6.8.0
	i4i.12xlarge	emr-5.36.1, emr-6.10.0
	i4i.16xlarge	emr-5.36.1, emr-6.8.0
	i4i.24xlarge	emr-5.36.1, emr-6.10.0
	i4i.32xlarge	emr-5.36.1, emr-6.8.0
	im4gn.xlarge	emr-5.35.0, emr-6.6.0, emr-7.0.0
	im4gn.2xlarge	emr-5.35.0, emr-6.6.0, emr-7.0.0
	im4gn.4xlarge	emr-5.35.0, emr-6.6.0, emr-7.0.0
	im4gn.8xlarge	emr-5.35.0, emr-6.6.0, emr-7.0.0
	im4gn.16xlarge	emr-5.35.0, emr-6.6.0, emr-7.0.0
	is4gen.xlarge	emr-5.35.0, emr-6.6.0, emr-7.0.0
	is4gen.2xlarge	emr-5.35.0, emr-6.6.0, emr-7.0.0
	is4gen.4xlarge	emr-5.35.0, emr-6.6.0, emr-7.0.0
	is4gen.8xlarge	emr-5.35.0, emr-6.6.0, emr-7.0.0

Canada (Central) - ca-central-1

Kelas Instans	Tipe instans	Versi EMR Amazon yang didukung minimum (5.x, 6.x, 7.x)
Tujuan Umum	m5.xlarge	emr-5.13.0, emr-6.0.0, emr-7.0.0
	m5.2xlarge	emr-5.13.0, emr-6.0.0, emr-7.0.0
	m5.4xlarge	emr-5.13.0, emr-6.0.0, emr-7.0.0
	m5.8xlarge	emr-5.13.0, emr-6.0.0, emr-7.0.0
	m5.12xlarge	emr-5.13.0, emr-6.0.0, emr-7.0.0
	m5.16xlarge	emr-5.13.0, emr-6.0.0, emr-7.0.0
	m5.24xlarge	emr-5.13.0, emr-6.0.0, emr-7.0.0
	m5a.xlarge	emr-5.20.0, emr-6.0.0, emr-7.0.0
	m5a.2xlarge	emr-5.20.0, emr-6.0.0, emr-7.0.0
	m5a.4xlarge	emr-5.20.0, emr-6.0.0, emr-7.0.0
	m5a.8xlarge	emr-5.20.0, emr-6.0.0, emr-7.0.0

Kelas Instans	Tipe instans	Versi EMR Amazon yang didukung minimum (5.x, 6.x, 7.x)
	m5a.12xlarge	emr-5.20.0, emr-6.0.0, emr-7.0.0
	m5a.16xlarge	emr-5.20.0, emr-6.0.0, emr-7.0.0
	m5a.24xlarge	emr-5.20.0, emr-6.0.0, emr-7.0.0
	m5ad.xlarge	emr-5.20.0, emr-6.0.0, emr-7.0.0
	m5ad.2xlarge	emr-5.20.0, emr-6.0.0, emr-7.0.0
	m5ad.4xlarge	emr-5.20.0, emr-6.0.0, emr-7.0.0
	m5ad.8xlarge	emr-5.33.0, emr-6.3.0, emr-7.0.0
	m5ad.12xlarge	emr-5.20.0, emr-6.0.0, emr-7.0.0
	m5ad.16xlarge	emr-5.33.0, emr-6.3.0, emr-7.0.0
	m5ad.24xlarge	emr-5.20.0, emr-6.0.0, emr-7.0.0
	m5d.xlarge	emr-5.13.0, emr-6.0.0, emr-7.0.0
	m5d.2xlarge	emr-5.13.0, emr-6.0.0, emr-7.0.0

Kelas Instans	Tipe instans	Versi EMR Amazon yang didukung minimum (5.x, 6.x, 7.x)
	m5d.4xlarge	emr-5.13.0, emr-6.0.0, emr-7.0.0
	m5d.8xlarge	emr-5.13.0, emr-6.0.0, emr-7.0.0
	m5d.12xlarge	emr-5.13.0, emr-6.0.0, emr-7.0.0
	m5d.16xlarge	emr-5.13.0, emr-6.0.0, emr-7.0.0
	m5d.24xlarge	emr-5.13.0, emr-6.0.0, emr-7.0.0
	m6a.xlarge	emr-5.36.1, emr-6.8.0, emr-7.0.0
	m6a.2xlarge	emr-5.36.1, emr-6.8.0, emr-7.0.0
	m6a.4xlarge	emr-5.36.1, emr-6.8.0, emr-7.0.0
	m6a.8xlarge	emr-5.36.1, emr-6.8.0, emr-7.0.0
	m6a.12xlarge	emr-5.36.1, emr-6.8.0, emr-7.0.0
	m6a.16xlarge	emr-5.36.1, emr-6.8.0, emr-7.0.0
	m6a.24xlarge	emr-5.36.1, emr-6.8.0, emr-7.0.0

Kelas Instans	Tipe instans	Versi EMR Amazon yang didukung minimum (5.x, 6.x, 7.x)
	m6a.32xlarge	emr-5.36.1, emr-6.8.0, emr-7.0.0
	m6a.48xlarge	emr-5.36.1, emr-6.8.0, emr-7.0.0
	m6g.xlarge	emr-5.30.0, emr-6.1.0, emr-7.0.0
	m6g.2xlarge	emr-5.30.0, emr-6.1.0, emr-7.0.0
	m6g.4xlarge	emr-5.30.0, emr-6.1.0, emr-7.0.0
	m6g.8xlarge	emr-5.30.0, emr-6.1.0, emr-7.0.0
	m6g.12xlarge	emr-5.30.0, emr-6.1.0, emr-7.0.0
	m6g.16xlarge	emr-5.30.0, emr-6.1.0, emr-7.0.0
	m6gd.xlarge	emr-5.33.0, emr-6.3.0, emr-7.0.0
	m6gd.2xlarge	emr-5.33.0, emr-6.3.0, emr-7.0.0
	m6gd.4xlarge	emr-5.33.0, emr-6.3.0, emr-7.0.0
	m6gd.8xlarge	emr-5.33.0, emr-6.3.0, emr-7.0.0

Kelas Instans	Tipe instans	Versi EMR Amazon yang didukung minimum (5.x, 6.x, 7.x)
	m6gd.12xlarge	emr-5.33.0, emr-6.3.0, emr-7.0.0
	m6gd.16xlarge	emr-5.33.0, emr-6.3.0, emr-7.0.0
	m6i.xlarge	emr-5.35.0, emr-6.6.0
	m6i.2xlarge	emr-5.35.0, emr-6.6.0
	m6i.4xlarge	emr-5.35.0, emr-6.6.0
	m6i.8xlarge	emr-5.35.0, emr-6.6.0
	m6i.12xlarge	emr-5.35.0, emr-6.6.0
	m6i.16xlarge	emr-5.35.0, emr-6.6.0
	m6i.24xlarge	emr-5.35.0, emr-6.6.0
	m6i.32xlarge	emr-5.35.0, emr-6.6.0
	m7g.xlarge	emr-5.36.1, emr-6.10.0, emr-7.0.0
	m7g.2xlarge	emr-5.36.1, emr-6.10.0, emr-7.0.0
	m7g.4xlarge	emr-5.36.1, emr-6.10.0, emr-7.0.0
	m7g.8xlarge	emr-5.36.1, emr-6.10.0, emr-7.0.0
	m7g.12xlarge	emr-5.36.1, emr-6.10.0, emr-7.0.0

Kelas Instans	Tipe instans	Versi EMR Amazon yang didukung minimum (5.x, 6.x, 7.x)
	m7g.16xlarge	emr-5.36.1, emr-6.10.0, emr-7.0.0
Komputasi Dioptimalkan	c5.xlarge	emr-5.13.0, emr-6.0.0, emr-7.0.0
	c5.2xlarge	emr-5.13.0, emr-6.0.0, emr-7.0.0
	c5.4xlarge	emr-5.13.0, emr-6.0.0, emr-7.0.0
	c5.9xlarge	emr-5.13.0, emr-6.0.0, emr-7.0.0
	c5.12xlarge	emr-5.13.0, emr-6.0.0, emr-7.0.0
	c5.18xlarge	emr-5.13.0, emr-6.0.0, emr-7.0.0
	c5.24xlarge	emr-5.13.0, emr-6.0.0, emr-7.0.0
	c5a.xlarge	emr-5.31.0, emr-6.1.0, emr-7.0.0
	c5a.2xlarge	emr-5.31.0, emr-6.1.0, emr-7.0.0
	c5a.4xlarge	emr-5.31.0, emr-6.1.0, emr-7.0.0
	c5a.8xlarge	emr-5.31.0, emr-6.1.0, emr-7.0.0

Kelas Instans	Tipe instans	Versi EMR Amazon yang didukung minimum (5.x, 6.x, 7.x)
	c5a.12xlarge	emr-5.31.0, emr-6.1.0, emr-7.0.0
	c5a.16xlarge	emr-5.31.0, emr-6.1.0, emr-7.0.0
	c5a.24xlarge	emr-5.31.0, emr-6.1.0, emr-7.0.0
	c5d.xlarge	emr-5.13.0, emr-6.0.0, emr-7.0.0
	c5d.2xlarge	emr-5.13.0, emr-6.0.0, emr-7.0.0
	c5d.4xlarge	emr-5.13.0, emr-6.0.0, emr-7.0.0
	c5d.9xlarge	emr-5.13.0, emr-6.0.0, emr-7.0.0
	c5d.12xlarge	emr-5.13.0, emr-6.0.0, emr-7.0.0
	c5d.18xlarge	emr-5.13.0, emr-6.0.0, emr-7.0.0
	c5d.24xlarge	emr-5.13.0, emr-6.0.0, emr-7.0.0
	c5n.xlarge	emr-5.20.0, emr-6.0.0, emr-7.0.0
	c5n.2xlarge	emr-5.20.0, emr-6.0.0, emr-7.0.0

Kelas Instans	Tipe instans	Versi EMR Amazon yang didukung minimum (5.x, 6.x, 7.x)
	c5n.4xlarge	emr-5.20.0, emr-6.0.0, emr-7.0.0
	c5n.9xlarge	emr-5.20.0, emr-6.0.0, emr-7.0.0
	c5n.18xlarge	emr-5.20.0, emr-6.0.0, emr-7.0.0
	c6a.xlarge	emr-5.36.1, emr-6.8.0, emr-7.0.0
	c6a.2xlarge	emr-5.36.1, emr-6.8.0, emr-7.0.0
	c6a.4xlarge	emr-5.36.1, emr-6.8.0, emr-7.0.0
	c6a.8xlarge	emr-5.36.1, emr-6.8.0, emr-7.0.0
	c6a.12xlarge	emr-5.36.1, emr-6.8.0, emr-7.0.0
	c6a.16xlarge	emr-5.36.1, emr-6.8.0, emr-7.0.0
	c6a.24xlarge	emr-5.36.1, emr-6.8.0, emr-7.0.0
	c6a.32xlarge	emr-5.36.1, emr-6.8.0, emr-7.0.0
	c6a.48xlarge	emr-5.36.1, emr-6.8.0, emr-7.0.0

Kelas Instans	Tipe instans	Versi EMR Amazon yang didukung minimum (5.x, 6.x, 7.x)
	c6g.xlarge	emr-5.31.0, emr-6.1.0, emr-7.0.0
	c6g.2xlarge	emr-5.31.0, emr-6.1.0, emr-7.0.0
	c6g.4xlarge	emr-5.31.0, emr-6.1.0, emr-7.0.0
	c6g.8xlarge	emr-5.31.0, emr-6.1.0, emr-7.0.0
	c6g.12xlarge	emr-5.31.0, emr-6.1.0, emr-7.0.0
	c6g.16xlarge	emr-5.31.0, emr-6.1.0, emr-7.0.0
	c6gd.xlarge	emr-5.33.0, emr-6.3.0, emr-7.0.0
	c6gd.2xlarge	emr-5.33.0, emr-6.3.0, emr-7.0.0
	c6gd.4xlarge	emr-5.33.0, emr-6.3.0, emr-7.0.0
	c6gd.8xlarge	emr-5.33.0, emr-6.3.0, emr-7.0.0
	c6gd.12xlarge	emr-5.33.0, emr-6.3.0, emr-7.0.0
	c6gd.16xlarge	emr-5.33.0, emr-6.3.0, emr-7.0.0

Kelas Instans	Tipe instans	Versi EMR Amazon yang didukung minimum (5.x, 6.x, 7.x)
	c6gn.xlarge	emr-5.33.0, emr-6.3.0, emr-7.0.0
	c6gn.2xlarge	emr-5.33.0, emr-6.3.0, emr-7.0.0
	c6gn.4xlarge	emr-5.33.0, emr-6.3.0, emr-7.0.0
	c6gn.8xlarge	emr-5.33.0, emr-6.3.0, emr-7.0.0
	c6gn.12xlarge	emr-5.33.0, emr-6.3.0, emr-7.0.0
	c6gn.16xlarge	emr-5.33.0, emr-6.3.0, emr-7.0.0
	c6i.xlarge	emr-5.35.0, emr-6.6.0
	c6i.2xlarge	emr-5.35.0, emr-6.6.0
	c6i.4xlarge	emr-5.35.0, emr-6.6.0
	c6i.8xlarge	emr-5.35.0, emr-6.6.0
	c6i.12xlarge	emr-5.35.0, emr-6.6.0
	c6i.16xlarge	emr-5.35.0, emr-6.6.0
	c6i.24xlarge	emr-5.35.0, emr-6.6.0
	c6i.32xlarge	emr-5.35.0, emr-6.6.0
	c6in.xlarge	emr-5.36.1, emr-6.10.0
	c6in.2xlarge	emr-5.36.1, emr-6.10.0

Kelas Instans	Tipe instans	Versi EMR Amazon yang didukung minimum (5.x, 6.x, 7.x)
	c6in.4xlarge	emr-5.36.1, emr-6.10.0
	c6in.8xlarge	emr-5.36.1, emr-6.10.0
	c6in.12xlarge	emr-5.36.1, emr-6.10.0
	c6in.16xlarge	emr-5.36.1, emr-6.10.0
	c6in.24xlarge	emr-5.36.1, emr-6.10.0
	c6in.32xlarge	emr-5.36.1, emr-6.10.0
	c7g.xlarge	emr-5.36.1, emr-6.7.0, emr-7.0.0
	c7g.2xlarge	emr-5.36.1, emr-6.7.0, emr-7.0.0
	c7g.4xlarge	emr-5.36.1, emr-6.7.0, emr-7.0.0
	c7g.8xlarge	emr-5.36.1, emr-6.7.0, emr-7.0.0
	c7g.12xlarge	emr-5.36.1, emr-6.7.0, emr-7.0.0
	c7g.16xlarge	emr-5.36.1, emr-6.7.0, emr-7.0.0
Komputasi yang Dipercepat	g3.4xlarge	emr-5.18.0, emr-6.0.0, emr-7.0.0
	g3.8xlarge	emr-5.18.0, emr-6.0.0, emr-7.0.0

Kelas Instans	Tipe instans	Versi EMR Amazon yang didukung minimum (5.x, 6.x, 7.x)
	g3.16xlarge	emr-5.18.0, emr-6.0.0, emr-7.0.0
	g4dn.xlarge	emr-5.30.0, emr-6.0.0, emr-7.0.0
	g4dn.2xlarge	emr-5.30.0, emr-6.0.0, emr-7.0.0
	g4dn.4xlarge	emr-5.30.0, emr-6.0.0, emr-7.0.0
	g4dn.8xlarge	emr-5.30.0, emr-6.0.0, emr-7.0.0
	g4dn.12xlarge	emr-5.30.0, emr-6.0.0, emr-7.0.0
	g4dn.16xlarge	emr-5.30.0, emr-6.0.0, emr-7.0.0
	g5.xlarge	emr-5.36.1, emr-6.9.0, emr-7.0.0
	g5.2xlarge	emr-5.36.1, emr-6.9.0, emr-7.0.0
	g5.4xlarge	emr-5.36.1, emr-6.9.0, emr-7.0.0
	g5.8xlarge	emr-5.36.1, emr-6.9.0, emr-7.0.0
	g5.12xlarge	emr-5.36.1, emr-6.9.0, emr-7.0.0

Kelas Instans	Tipe instans	Versi EMR Amazon yang didukung minimum (5.x, 6.x, 7.x)
	g5.16xlarge	emr-5.36.1, emr-6.9.0, emr-7.0.0
	g5.24xlarge	emr-5.36.1, emr-6.9.0, emr-7.0.0
	g5.48xlarge	emr-5.36.1, emr-6.9.0, emr-7.0.0
	p3.2xlarge	emr-5.10.0, emr-6.0.0, emr-7.0.0
	p3.8xlarge	emr-5.10.0, emr-6.0.0, emr-7.0.0
	p3.16xlarge	emr-5.10.0, emr-6.0.0, emr-7.0.0
Memori Dioptimalkan	r5.xlarge	emr-5.13.0, emr-6.0.0, emr-7.0.0
	r5.2xlarge	emr-5.13.0, emr-6.0.0, emr-7.0.0
	r5.4xlarge	emr-5.13.0, emr-6.0.0, emr-7.0.0
	r5.8xlarge	emr-5.13.0, emr-6.0.0, emr-7.0.0
	r5.12xlarge	emr-5.13.0, emr-6.0.0, emr-7.0.0
	r5.16xlarge	emr-5.13.0, emr-6.0.0, emr-7.0.0

Kelas Instans	Tipe instans	Versi EMR Amazon yang didukung minimum (5.x, 6.x, 7.x)
	r5.24xlarge	emr-5.13.0, emr-6.0.0, emr-7.0.0
	r5a.xlarge	emr-5.20.0, emr-6.0.0, emr-7.0.0
	r5a.2xlarge	emr-5.20.0, emr-6.0.0, emr-7.0.0
	r5a.4xlarge	emr-5.20.0, emr-6.0.0, emr-7.0.0
	r5a.8xlarge	emr-5.20.0, emr-6.0.0, emr-7.0.0
	r5a.12xlarge	emr-5.20.0, emr-6.0.0, emr-7.0.0
	r5a.16xlarge	emr-5.20.0, emr-6.0.0, emr-7.0.0
	r5a.24xlarge	emr-5.20.0, emr-6.0.0, emr-7.0.0
	r5ad.xlarge	emr-5.20.0, emr-6.0.0, emr-7.0.0
	r5ad.2xlarge	emr-5.20.0, emr-6.0.0, emr-7.0.0
	r5ad.4xlarge	emr-5.20.0, emr-6.0.0, emr-7.0.0
	r5ad.8xlarge	emr-5.33.0, emr-6.3.0, emr-7.0.0

Kelas Instans	Tipe instans	Versi EMR Amazon yang didukung minimum (5.x, 6.x, 7.x)
	r5ad.12xlarge	emr-5.20.0, emr-6.0.0, emr-7.0.0
	r5ad.16xlarge	emr-5.33.0, emr-6.3.0, emr-7.0.0
	r5ad.24xlarge	emr-5.20.0, emr-6.0.0, emr-7.0.0
	r5b.xlarge	emr-5.33.0, emr-6.3.0, emr-7.0.0
	r5b.2xlarge	emr-5.33.0, emr-6.3.0, emr-7.0.0
	r5b.4xlarge	emr-5.33.0, emr-6.3.0, emr-7.0.0
	r5b.8xlarge	emr-5.33.0, emr-6.3.0, emr-7.0.0
	r5b.12xlarge	emr-5.33.0, emr-6.3.0, emr-7.0.0
	r5b.16xlarge	emr-5.33.0, emr-6.3.0, emr-7.0.0
	r5b.24xlarge	emr-5.33.0, emr-6.3.0, emr-7.0.0
	r5d.xlarge	emr-5.13.0, emr-6.0.0, emr-7.0.0
	r5d.2xlarge	emr-5.13.0, emr-6.0.0, emr-7.0.0

Kelas Instans	Tipe instans	Versi EMR Amazon yang didukung minimum (5.x, 6.x, 7.x)
	r5d.4xlarge	emr-5.13.0, emr-6.0.0, emr-7.0.0
	r5d.8xlarge	emr-5.13.0, emr-6.0.0, emr-7.0.0
	r5d.12xlarge	emr-5.13.0, emr-6.0.0, emr-7.0.0
	r5d.16xlarge	emr-5.13.0, emr-6.0.0, emr-7.0.0
	r5d.24xlarge	emr-5.13.0, emr-6.0.0, emr-7.0.0
	r5n.xlarge	emr-5.31.0, emr-6.1.0, emr-7.0.0
	r5n.2xlarge	emr-5.31.0, emr-6.1.0, emr-7.0.0
	r5n.4xlarge	emr-5.31.0, emr-6.1.0, emr-7.0.0
	r5n.8xlarge	emr-5.31.0, emr-6.1.0, emr-7.0.0
	r5n.12xlarge	emr-5.31.0, emr-6.1.0, emr-7.0.0
	r5n.16xlarge	emr-5.31.0, emr-6.1.0, emr-7.0.0
	r5n.24xlarge	emr-5.31.0, emr-6.1.0, emr-7.0.0

Kelas Instans	Tipe instans	Versi EMR Amazon yang didukung minimum (5.x, 6.x, 7.x)
	r6g.xlarge	emr-5.31.0, emr-6.1.0, emr-7.0.0
	r6g.2xlarge	emr-5.31.0, emr-6.1.0, emr-7.0.0
	r6g.4xlarge	emr-5.31.0, emr-6.1.0, emr-7.0.0
	r6g.8xlarge	emr-5.31.0, emr-6.1.0, emr-7.0.0
	r6g.12xlarge	emr-5.31.0, emr-6.1.0, emr-7.0.0
	r6g.16xlarge	emr-5.31.0, emr-6.1.0, emr-7.0.0
	r6gd.xlarge	emr-5.33.0, emr-6.3.0, emr-7.0.0
	r6gd.2xlarge	emr-5.33.0, emr-6.3.0, emr-7.0.0
	r6gd.4xlarge	emr-5.33.0, emr-6.3.0, emr-7.0.0
	r6gd.8xlarge	emr-5.33.0, emr-6.3.0, emr-7.0.0
	r6gd.12xlarge	emr-5.33.0, emr-6.3.0, emr-7.0.0
	r6gd.16xlarge	emr-5.33.0, emr-6.3.0, emr-7.0.0

Kelas Instans	Tipe instans	Versi EMR Amazon yang didukung minimum (5.x, 6.x, 7.x)
	r6i.xlarge	emr-5.35.0, emr-6.6.0
	r6i.2xlarge	emr-5.35.0, emr-6.6.0
	r6i.4xlarge	emr-5.35.0, emr-6.6.0
	r6i.8xlarge	emr-5.35.0, emr-6.6.0
	r6i.12xlarge	emr-5.35.0, emr-6.6.0
	r6i.16xlarge	emr-5.35.0, emr-6.6.0
	r6i.24xlarge	emr-5.35.0, emr-6.6.0
	r6i.32xlarge	emr-5.35.0, emr-6.6.0
	r7g.xlarge	emr-5.36.1, emr-6.10.0, emr-7.0.0
	r7g.2xlarge	emr-5.36.1, emr-6.10.0, emr-7.0.0
	r7g.4xlarge	emr-5.36.1, emr-6.10.0, emr-7.0.0
	r7g.8xlarge	emr-5.36.1, emr-6.10.0, emr-7.0.0
	r7g.12xlarge	emr-5.36.1, emr-6.10.0, emr-7.0.0
	r7g.16xlarge	emr-5.36.1, emr-6.10.0, emr-7.0.0
	x1.16xlarge	emr-5.36.1, emr-6.9.0, emr-7.0.0

Kelas Instans	Tipe instans	Versi EMR Amazon yang didukung minimum (5.x, 6.x, 7.x)
	x1.32xlarge	emr-5.36.1, emr-6.9.0, emr-7.0.0
	x1e.xlarge	emr-5.36.1, emr-6.10.0, emr-7.0.0
	x1e.2xlarge	emr-5.36.1, emr-6.10.0, emr-7.0.0
	x1e.4xlarge	emr-5.36.1, emr-6.10.0, emr-7.0.0
	x1e.8xlarge	emr-5.36.1, emr-6.10.0, emr-7.0.0
	x1e.16xlarge	emr-5.36.1, emr-6.10.0, emr-7.0.0
	x1e.32xlarge	emr-5.36.1, emr-6.10.0, emr-7.0.0
	x2idn.16xlarge	emr-5.36.1, emr-6.9.0
	x2idn.24xlarge	emr-5.36.1, emr-6.9.0
	x2idn.32xlarge	emr-5.36.1, emr-6.9.0
	x2iedn.xlarge	emr-5.36.1, emr-6.9.0
	x2iedn.2xlarge	emr-5.36.1, emr-6.9.0
	x2iedn.4xlarge	emr-5.36.1, emr-6.9.0
	x2iedn.8xlarge	emr-5.36.1, emr-6.9.0
	x2iedn.16xlarge	emr-5.36.1, emr-6.9.0

Kelas Instans	Tipe instans	Versi EMR Amazon yang didukung minimum (5.x, 6.x, 7.x)
	x2iedn.24xlarge	emr-5.36.1, emr-6.9.0
	x2iedn.32xlarge	emr-5.36.1, emr-6.9.0
Penyimpanan Dioptimalkan	d3.xlarge	emr-5.33.0, emr-6.3.0, emr-7.0.0
	d3.2xlarge	emr-5.33.0, emr-6.3.0, emr-7.0.0
	d3.4xlarge	emr-5.33.0, emr-6.3.0, emr-7.0.0
	d3.8xlarge	emr-5.33.0, emr-6.3.0, emr-7.0.0
	i3.xlarge	emr-5.9.0, emr-6.0.0, emr-7.0.0
	i3.2xlarge	emr-5.9.0, emr-6.0.0, emr-7.0.0
	i3.4xlarge	emr-5.9.0, emr-6.0.0, emr-7.0.0
	i3.8xlarge	emr-5.9.0, emr-6.0.0, emr-7.0.0
	i3.16xlarge	emr-5.9.0, emr-6.0.0, emr-7.0.0
	i3en.xlarge	emr-5.25.0, emr-6.0.0, emr-7.0.0
	i3en.2xlarge	emr-5.25.0, emr-6.0.0, emr-7.0.0

Kelas Instans	Tipe instans	Versi EMR Amazon yang didukung minimum (5.x, 6.x, 7.x)
	i3en.3xlarge	emr-5.25.0, emr-6.0.0, emr-7.0.0
	i3en.6xlarge	emr-5.25.0, emr-6.0.0, emr-7.0.0
	i3en.12xlarge	emr-5.25.0, emr-6.0.0, emr-7.0.0
	i3en.24xlarge	emr-5.25.0, emr-6.0.0, emr-7.0.0
	i4g.xlarge	emr-5.36.1, emr-6.10.0, emr-7.0.0
	i4g.2xlarge	emr-5.36.1, emr-6.10.0, emr-7.0.0
	i4g.4xlarge	emr-5.36.1, emr-6.10.0, emr-7.0.0
	i4g.8xlarge	emr-5.36.1, emr-6.10.0, emr-7.0.0
	i4g.16xlarge	emr-5.36.1, emr-6.10.0, emr-7.0.0
	i4i.xlarge	emr-5.36.1, emr-6.8.0
	i4i.2xlarge	emr-5.36.1, emr-6.8.0
	i4i.4xlarge	emr-5.36.1, emr-6.8.0
	i4i.8xlarge	emr-5.36.1, emr-6.8.0
	i4i.12xlarge	emr-5.36.1, emr-6.10.0

Kelas Instans	Tipe instans	Versi EMR Amazon yang didukung minimum (5.x, 6.x, 7.x)
	i4i.16xlarge	emr-5.36.1, emr-6.8.0
	i4i.24xlarge	emr-5.36.1, emr-6.10.0
	i4i.32xlarge	emr-5.36.1, emr-6.8.0
	im4gn.xlarge	emr-5.35.0, emr-6.6.0, emr-7.0.0
	im4gn.2xlarge	emr-5.35.0, emr-6.6.0, emr-7.0.0
	im4gn.4xlarge	emr-5.35.0, emr-6.6.0, emr-7.0.0
	im4gn.8xlarge	emr-5.35.0, emr-6.6.0, emr-7.0.0
	im4gn.16xlarge	emr-5.35.0, emr-6.6.0, emr-7.0.0
	is4gen.xlarge	emr-5.35.0, emr-6.6.0, emr-7.0.0
	is4gen.2xlarge	emr-5.35.0, emr-6.6.0, emr-7.0.0
	is4gen.4xlarge	emr-5.35.0, emr-6.6.0, emr-7.0.0
	is4gen.8xlarge	emr-5.35.0, emr-6.6.0, emr-7.0.0

Kanada Barat (Calgary) - ca-barat-1

Kelas Instans	Tipe instans	Versi EMR Amazon yang didukung minimum (5.x, 6.x, 7.x)
Tujuan Umum	m5.xlarge	emr-5.36.1, emr-6.9.0, emr-7.0.0
	m5.2xlarge	emr-5.36.1, emr-6.9.0, emr-7.0.0
	m5.4xlarge	emr-5.36.1, emr-6.9.0, emr-7.0.0
	m5.8xlarge	emr-5.36.1, emr-6.9.0, emr-7.0.0
	m5.12xlarge	emr-5.36.1, emr-6.9.0, emr-7.0.0
	m5.16xlarge	emr-5.36.1, emr-6.9.0, emr-7.0.0
	m5.24xlarge	emr-5.36.1, emr-6.9.0, emr-7.0.0
	m5d.xlarge	emr-5.36.1, emr-6.9.0, emr-7.0.0
	m5d.2xlarge	emr-5.36.1, emr-6.9.0, emr-7.0.0
	m5d.4xlarge	emr-5.36.1, emr-6.9.0, emr-7.0.0
	m5d.8xlarge	emr-5.36.1, emr-6.9.0, emr-7.0.0

Kelas Instans	Tipe instans	Versi EMR Amazon yang didukung minimum (5.x, 6.x, 7.x)
	m5d.12xlarge	emr-5.36.1, emr-6.9.0, emr-7.0.0
	m5d.16xlarge	emr-5.36.1, emr-6.9.0, emr-7.0.0
	m5d.24xlarge	emr-5.36.1, emr-6.9.0, emr-7.0.0
	m6g.xlarge	emr-5.36.1, emr-6.9.0, emr-7.0.0
	m6g.2xlarge	emr-5.36.1, emr-6.9.0, emr-7.0.0
	m6g.4xlarge	emr-5.36.1, emr-6.9.0, emr-7.0.0
	m6g.8xlarge	emr-5.36.1, emr-6.9.0, emr-7.0.0
	m6g.12xlarge	emr-5.36.1, emr-6.9.0, emr-7.0.0
	m6g.16xlarge	emr-5.36.1, emr-6.9.0, emr-7.0.0
	m6gd.xlarge	emr-5.36.1, emr-6.9.0, emr-7.0.0
	m6gd.2xlarge	emr-5.36.1, emr-6.9.0, emr-7.0.0
	m6gd.4xlarge	emr-5.36.1, emr-6.9.0, emr-7.0.0

Kelas Instans	Tipe instans	Versi EMR Amazon yang didukung minimum (5.x, 6.x, 7.x)
	m6gd.8xlarge	emr-5.36.1, emr-6.9.0, emr-7.0.0
	m6gd.12xlarge	emr-5.36.1, emr-6.9.0, emr-7.0.0
	m6gd.16xlarge	emr-5.36.1, emr-6.9.0, emr-7.0.0
	m6i.xlarge	emr-5.36.1, emr-6.9.0
	m6i.2xlarge	emr-5.36.1, emr-6.9.0
	m6i.4xlarge	emr-5.36.1, emr-6.9.0
	m6i.8xlarge	emr-5.36.1, emr-6.9.0
	m6i.12xlarge	emr-5.36.1, emr-6.9.0
	m6i.16xlarge	emr-5.36.1, emr-6.9.0
	m6i.24xlarge	emr-5.36.1, emr-6.9.0
	m6i.32xlarge	emr-5.36.1, emr-6.9.0
	m6id.xlarge	emr-5.36.1, emr-6.9.0
	m6id.2xlarge	emr-5.36.1, emr-6.9.0
	m6id.4xlarge	emr-5.36.1, emr-6.9.0
	m6id.8xlarge	emr-5.36.1, emr-6.9.0
	m6id.12xlarge	emr-5.36.1, emr-6.9.0
	m6id.16xlarge	emr-5.36.1, emr-6.9.0

Kelas Instans	Tipe instans	Versi EMR Amazon yang didukung minimum (5.x, 6.x, 7.x)
	m6id.24xlarge	emr-5.36.1, emr-6.9.0
	m6id.32xlarge	emr-5.36.1, emr-6.9.0
Komputasi Dioptimalkan	c5.xlarge	emr-5.36.1, emr-6.9.0, emr-7.0.0
	c5.2xlarge	emr-5.36.1, emr-6.9.0, emr-7.0.0
	c5.4xlarge	emr-5.36.1, emr-6.9.0, emr-7.0.0
	c5.9xlarge	emr-5.36.1, emr-6.9.0, emr-7.0.0
	c5.12xlarge	emr-5.36.1, emr-6.9.0, emr-7.0.0
	c5.18xlarge	emr-5.36.1, emr-6.9.0, emr-7.0.0
	c5.24xlarge	emr-5.36.1, emr-6.9.0, emr-7.0.0
	c6g.xlarge	emr-5.36.1, emr-6.9.0, emr-7.0.0
	c6g.2xlarge	emr-5.36.1, emr-6.9.0, emr-7.0.0
	c6g.4xlarge	emr-5.36.1, emr-6.9.0, emr-7.0.0
	c6g.8xlarge	emr-5.36.1, emr-6.9.0, emr-7.0.0

Kelas Instans	Tipe instans	Versi EMR Amazon yang didukung minimum (5.x, 6.x, 7.x)
	c6g.12xlarge	emr-5.36.1, emr-6.9.0, emr-7.0.0
	c6g.16xlarge	emr-5.36.1, emr-6.9.0, emr-7.0.0
	c6gn.xlarge	emr-5.36.1, emr-6.9.0, emr-7.0.0
	c6gn.2xlarge	emr-5.36.1, emr-6.9.0, emr-7.0.0
	c6gn.4xlarge	emr-5.36.1, emr-6.9.0, emr-7.0.0
	c6gn.8xlarge	emr-5.36.1, emr-6.9.0, emr-7.0.0
	c6gn.12xlarge	emr-5.36.1, emr-6.9.0, emr-7.0.0
	c6gn.16xlarge	emr-5.36.1, emr-6.9.0, emr-7.0.0
	c6i.xlarge	emr-5.36.1, emr-6.9.0
	c6i.2xlarge	emr-5.36.1, emr-6.9.0
	c6i.4xlarge	emr-5.36.1, emr-6.9.0
	c6i.8xlarge	emr-5.36.1, emr-6.9.0
	c6i.12xlarge	emr-5.36.1, emr-6.9.0
	c6i.16xlarge	emr-5.36.1, emr-6.9.0

Kelas Instans	Tipe instans	Versi EMR Amazon yang didukung minimum (5.x, 6.x, 7.x)
	c6i.24xlarge	emr-5.36.1, emr-6.9.0
	c6i.32xlarge	emr-5.36.1, emr-6.9.0
	c6id.xlarge	emr-5.36.1, emr-6.9.0
	c6id.2xlarge	emr-5.36.1, emr-6.9.0
	c6id.4xlarge	emr-5.36.1, emr-6.9.0
	c6id.8xlarge	emr-5.36.1, emr-6.9.0
	c6id.12xlarge	emr-5.36.1, emr-6.9.0
	c6id.16xlarge	emr-5.36.1, emr-6.9.0
	c6id.24xlarge	emr-5.36.1, emr-6.9.0
	c6id.32xlarge	emr-5.36.1, emr-6.9.0
Memori Dioptimalkan	r5.xlarge	emr-5.36.1, emr-6.9.0, emr-7.0.0
	r5.2xlarge	emr-5.36.1, emr-6.9.0, emr-7.0.0
	r5.4xlarge	emr-5.36.1, emr-6.9.0, emr-7.0.0
	r5.8xlarge	emr-5.36.1, emr-6.9.0, emr-7.0.0
	r5.12xlarge	emr-5.36.1, emr-6.9.0, emr-7.0.0
	r5.16xlarge	emr-5.36.1, emr-6.9.0, emr-7.0.0

Kelas Instans	Tipe instans	Versi EMR Amazon yang didukung minimum (5.x, 6.x, 7.x)
	r5.24xlarge	emr-5.36.1, emr-6.9.0, emr-7.0.0
	r6g.xlarge	emr-5.36.1, emr-6.9.0, emr-7.0.0
	r6g.2xlarge	emr-5.36.1, emr-6.9.0, emr-7.0.0
	r6g.4xlarge	emr-5.36.1, emr-6.9.0, emr-7.0.0
	r6g.8xlarge	emr-5.36.1, emr-6.9.0, emr-7.0.0
	r6g.12xlarge	emr-5.36.1, emr-6.9.0, emr-7.0.0
	r6g.16xlarge	emr-5.36.1, emr-6.9.0, emr-7.0.0
	r6i.xlarge	emr-5.36.1, emr-6.9.0
	r6i.2xlarge	emr-5.36.1, emr-6.9.0
	r6i.4xlarge	emr-5.36.1, emr-6.9.0
	r6i.8xlarge	emr-5.36.1, emr-6.9.0
	r6i.12xlarge	emr-5.36.1, emr-6.9.0
	r6i.16xlarge	emr-5.36.1, emr-6.9.0
	r6i.24xlarge	emr-5.36.1, emr-6.9.0
	r6i.32xlarge	emr-5.36.1, emr-6.9.0

Kelas Instans	Tipe instans	Versi EMR Amazon yang didukung minimum (5.x, 6.x, 7.x)
	r6id.xlarge	emr-5.36.1, emr-6.9.0
	r6id.2xlarge	emr-5.36.1, emr-6.9.0
	r6id.4xlarge	emr-5.36.1, emr-6.9.0
	r6id.8xlarge	emr-5.36.1, emr-6.9.0
	r6id.12xlarge	emr-5.36.1, emr-6.9.0
	r6id.16xlarge	emr-5.36.1, emr-6.9.0
	r6id.24xlarge	emr-5.36.1, emr-6.9.0
	r6id.32xlarge	emr-5.36.1, emr-6.9.0
Penyimpanan Dioptimalkan	i3en.xlarge	emr-5.36.1, emr-6.9.0, emr-7.0.0
	i3en.2xlarge	emr-5.36.1, emr-6.9.0, emr-7.0.0
	i3en.3xlarge	emr-5.36.1, emr-6.9.0, emr-7.0.0
	i3en.6xlarge	emr-5.36.1, emr-6.9.0, emr-7.0.0
	i3en.12xlarge	emr-5.36.1, emr-6.9.0, emr-7.0.0
	i3en.24xlarge	emr-5.36.1, emr-6.9.0, emr-7.0.0
	i4i.xlarge	emr-5.36.1, emr-6.9.0
	i4i.2xlarge	emr-5.36.1, emr-6.9.0

Kelas Instans	Tipe instans	Versi EMR Amazon yang didukung minimum (5.x, 6.x, 7.x)
	i4i.4xlarge	emr-5.36.1, emr-6.9.0
	i4i.8xlarge	emr-5.36.1, emr-6.9.0
	i4i.16xlarge	emr-5.36.1, emr-6.9.0
	i4i.32xlarge	emr-5.36.1, emr-6.9.0

Tiongkok (Ningxia) - cn-barat laut-1

Kelas Instans	Tipe instans	Versi EMR Amazon yang didukung minimum (5.x, 6.x, 7.x)
Tujuan Umum	m5.xlarge	emr-5.13.0, emr-6.0.0, emr-7.0.0
	m5.2xlarge	emr-5.13.0, emr-6.0.0, emr-7.0.0
	m5.4xlarge	emr-5.13.0, emr-6.0.0, emr-7.0.0
	m5.8xlarge	emr-5.13.0, emr-6.0.0, emr-7.0.0
	m5.12xlarge	emr-5.13.0, emr-6.0.0, emr-7.0.0
	m5.16xlarge	emr-5.13.0, emr-6.0.0, emr-7.0.0
	m5.24xlarge	emr-5.13.0, emr-6.0.0, emr-7.0.0

Kelas Instans	Tipe instans	Versi EMR Amazon yang didukung minimum (5.x, 6.x, 7.x)
	m5a.xlarge	emr-5.20.0, emr-6.0.0, emr-7.0.0
	m5a.2xlarge	emr-5.20.0, emr-6.0.0, emr-7.0.0
	m5a.4xlarge	emr-5.20.0, emr-6.0.0, emr-7.0.0
	m5a.8xlarge	emr-5.20.0, emr-6.0.0, emr-7.0.0
	m5a.12xlarge	emr-5.20.0, emr-6.0.0, emr-7.0.0
	m5a.16xlarge	emr-5.20.0, emr-6.0.0, emr-7.0.0
	m5a.24xlarge	emr-5.20.0, emr-6.0.0, emr-7.0.0
	m5d.xlarge	emr-5.13.0, emr-6.0.0, emr-7.0.0
	m5d.2xlarge	emr-5.13.0, emr-6.0.0, emr-7.0.0
	m5d.4xlarge	emr-5.13.0, emr-6.0.0, emr-7.0.0
	m5d.8xlarge	emr-5.13.0, emr-6.0.0, emr-7.0.0
	m5d.12xlarge	emr-5.13.0, emr-6.0.0, emr-7.0.0

Kelas Instans	Tipe instans	Versi EMR Amazon yang didukung minimum (5.x, 6.x, 7.x)
	m5d.16xlarge	emr-5.13.0, emr-6.0.0, emr-7.0.0
	m5d.24xlarge	emr-5.13.0, emr-6.0.0, emr-7.0.0
	m6g.xlarge	emr-5.30.0, emr-6.1.0, emr-7.0.0
	m6g.2xlarge	emr-5.30.0, emr-6.1.0, emr-7.0.0
	m6g.4xlarge	emr-5.30.0, emr-6.1.0, emr-7.0.0
	m6g.8xlarge	emr-5.30.0, emr-6.1.0, emr-7.0.0
	m6g.12xlarge	emr-5.30.0, emr-6.1.0, emr-7.0.0
	m6g.16xlarge	emr-5.30.0, emr-6.1.0, emr-7.0.0
	m6i.xlarge	emr-5.35.0, emr-6.6.0
	m6i.2xlarge	emr-5.35.0, emr-6.6.0
	m6i.4xlarge	emr-5.35.0, emr-6.6.0
	m6i.8xlarge	emr-5.35.0, emr-6.6.0
	m6i.12xlarge	emr-5.35.0, emr-6.6.0
	m6i.16xlarge	emr-5.35.0, emr-6.6.0

Kelas Instans	Tipe instans	Versi EMR Amazon yang didukung minimum (5.x, 6.x, 7.x)
	m6i.24xlarge	emr-5.35.0, emr-6.6.0
	m6i.32xlarge	emr-5.35.0, emr-6.6.0
	m7g.xlarge	emr-5.36.1, emr-6.10.0, emr-7.0.0
	m7g.2xlarge	emr-5.36.1, emr-6.10.0, emr-7.0.0
	m7g.4xlarge	emr-5.36.1, emr-6.10.0, emr-7.0.0
	m7g.8xlarge	emr-5.36.1, emr-6.10.0, emr-7.0.0
	m7g.12xlarge	emr-5.36.1, emr-6.10.0, emr-7.0.0
	m7g.16xlarge	emr-5.36.1, emr-6.10.0, emr-7.0.0
Komputasi Dioptimalkan	c5.xlarge	emr-5.13.0, emr-6.0.0, emr-7.0.0
	c5.2xlarge	emr-5.13.0, emr-6.0.0, emr-7.0.0
	c5.4xlarge	emr-5.13.0, emr-6.0.0, emr-7.0.0
	c5.9xlarge	emr-5.13.0, emr-6.0.0, emr-7.0.0
	c5.12xlarge	emr-5.13.0, emr-6.0.0, emr-7.0.0

Kelas Instans	Tipe instans	Versi EMR Amazon yang didukung minimum (5.x, 6.x, 7.x)
	c5.18xlarge	emr-5.13.0, emr-6.0.0, emr-7.0.0
	c5.24xlarge	emr-5.13.0, emr-6.0.0, emr-7.0.0
	c5a.xlarge	emr-5.31.0, emr-6.1.0, emr-7.0.0
	c5a.2xlarge	emr-5.31.0, emr-6.1.0, emr-7.0.0
	c5a.4xlarge	emr-5.31.0, emr-6.1.0, emr-7.0.0
	c5a.8xlarge	emr-5.31.0, emr-6.1.0, emr-7.0.0
	c5a.12xlarge	emr-5.31.0, emr-6.1.0, emr-7.0.0
	c5a.16xlarge	emr-5.31.0, emr-6.1.0, emr-7.0.0
	c5a.24xlarge	emr-5.31.0, emr-6.1.0, emr-7.0.0
	c5d.xlarge	emr-5.13.0, emr-6.0.0, emr-7.0.0
	c5d.2xlarge	emr-5.13.0, emr-6.0.0, emr-7.0.0
	c5d.4xlarge	emr-5.13.0, emr-6.0.0, emr-7.0.0

Kelas Instans	Tipe instans	Versi EMR Amazon yang didukung minimum (5.x, 6.x, 7.x)
	c5d.9xlarge	emr-5.13.0, emr-6.0.0, emr-7.0.0
	c5d.18xlarge	emr-5.13.0, emr-6.0.0, emr-7.0.0
	c6g.xlarge	emr-5.31.0, emr-6.1.0, emr-7.0.0
	c6g.2xlarge	emr-5.31.0, emr-6.1.0, emr-7.0.0
	c6g.4xlarge	emr-5.31.0, emr-6.1.0, emr-7.0.0
	c6g.8xlarge	emr-5.31.0, emr-6.1.0, emr-7.0.0
	c6g.12xlarge	emr-5.31.0, emr-6.1.0, emr-7.0.0
	c6g.16xlarge	emr-5.31.0, emr-6.1.0, emr-7.0.0
	c6i.xlarge	emr-5.35.0, emr-6.6.0
	c6i.2xlarge	emr-5.35.0, emr-6.6.0
	c6i.4xlarge	emr-5.35.0, emr-6.6.0
	c6i.8xlarge	emr-5.35.0, emr-6.6.0
	c6i.12xlarge	emr-5.35.0, emr-6.6.0
	c6i.16xlarge	emr-5.35.0, emr-6.6.0

Kelas Instans	Tipe instans	Versi EMR Amazon yang didukung minimum (5.x, 6.x, 7.x)
	c6i.24xlarge	emr-5.35.0, emr-6.6.0
	c6i.32xlarge	emr-5.35.0, emr-6.6.0
	c6in.xlarge	emr-5.36.1, emr-6.10.0
	c6in.2xlarge	emr-5.36.1, emr-6.10.0
	c6in.4xlarge	emr-5.36.1, emr-6.10.0
	c6in.8xlarge	emr-5.36.1, emr-6.10.0
	c6in.12xlarge	emr-5.36.1, emr-6.10.0
	c6in.16xlarge	emr-5.36.1, emr-6.10.0
	c6in.24xlarge	emr-5.36.1, emr-6.10.0
	c6in.32xlarge	emr-5.36.1, emr-6.10.0
	c7g.xlarge	emr-5.36.1, emr-6.7.0, emr-7.0.0
	c7g.2xlarge	emr-5.36.1, emr-6.7.0, emr-7.0.0
	c7g.4xlarge	emr-5.36.1, emr-6.7.0, emr-7.0.0
	c7g.8xlarge	emr-5.36.1, emr-6.7.0, emr-7.0.0
	c7g.12xlarge	emr-5.36.1, emr-6.7.0, emr-7.0.0
	c7g.16xlarge	emr-5.36.1, emr-6.7.0, emr-7.0.0

Kelas Instans	Tipe instans	Versi EMR Amazon yang didukung minimum (5.x, 6.x, 7.x)
Komputasi yang Dipercepat	g4dn.xlarge	emr-5.30.0, emr-6.0.0, emr-7.0.0
	g4dn.2xlarge	emr-5.30.0, emr-6.0.0, emr-7.0.0
	g4dn.4xlarge	emr-5.30.0, emr-6.0.0, emr-7.0.0
	g4dn.8xlarge	emr-5.30.0, emr-6.0.0, emr-7.0.0
	g4dn.12xlarge	emr-5.30.0, emr-6.0.0, emr-7.0.0
	g4dn.16xlarge	emr-5.30.0, emr-6.0.0, emr-7.0.0
	p3.2xlarge	emr-5.10.0, emr-6.0.0, emr-7.0.0
	p3.8xlarge	emr-5.10.0, emr-6.0.0, emr-7.0.0
	p3.16xlarge	emr-5.10.0, emr-6.0.0, emr-7.0.0
Memori Dioptimalkan	r5.xlarge	emr-5.13.0, emr-6.0.0, emr-7.0.0
	r5.2xlarge	emr-5.13.0, emr-6.0.0, emr-7.0.0
	r5.4xlarge	emr-5.13.0, emr-6.0.0, emr-7.0.0

Kelas Instans	Tipe instans	Versi EMR Amazon yang didukung minimum (5.x, 6.x, 7.x)
	r5.8xlarge	emr-5.13.0, emr-6.0.0, emr-7.0.0
	r5.12xlarge	emr-5.13.0, emr-6.0.0, emr-7.0.0
	r5.16xlarge	emr-5.13.0, emr-6.0.0, emr-7.0.0
	r5.24xlarge	emr-5.13.0, emr-6.0.0, emr-7.0.0
	r5a.xlarge	emr-5.20.0, emr-6.0.0, emr-7.0.0
	r5a.2xlarge	emr-5.20.0, emr-6.0.0, emr-7.0.0
	r5a.4xlarge	emr-5.20.0, emr-6.0.0, emr-7.0.0
	r5a.8xlarge	emr-5.20.0, emr-6.0.0, emr-7.0.0
	r5a.12xlarge	emr-5.20.0, emr-6.0.0, emr-7.0.0
	r5a.16xlarge	emr-5.20.0, emr-6.0.0, emr-7.0.0
	r5a.24xlarge	emr-5.20.0, emr-6.0.0, emr-7.0.0
	r5d.xlarge	emr-5.13.0, emr-6.0.0, emr-7.0.0

Kelas Instans	Tipe instans	Versi EMR Amazon yang didukung minimum (5.x, 6.x, 7.x)
	r5d.2xlarge	emr-5.13.0, emr-6.0.0, emr-7.0.0
	r5d.4xlarge	emr-5.13.0, emr-6.0.0, emr-7.0.0
	r5d.8xlarge	emr-5.13.0, emr-6.0.0, emr-7.0.0
	r5d.12xlarge	emr-5.13.0, emr-6.0.0, emr-7.0.0
	r5d.16xlarge	emr-5.13.0, emr-6.0.0, emr-7.0.0
	r5d.24xlarge	emr-5.13.0, emr-6.0.0, emr-7.0.0
	r6g.xlarge	emr-5.31.0, emr-6.1.0, emr-7.0.0
	r6g.2xlarge	emr-5.31.0, emr-6.1.0, emr-7.0.0
	r6g.4xlarge	emr-5.31.0, emr-6.1.0, emr-7.0.0
	r6g.8xlarge	emr-5.31.0, emr-6.1.0, emr-7.0.0
	r6g.12xlarge	emr-5.31.0, emr-6.1.0, emr-7.0.0
	r6g.16xlarge	emr-5.31.0, emr-6.1.0, emr-7.0.0

Kelas Instans	Tipe instans	Versi EMR Amazon yang didukung minimum (5.x, 6.x, 7.x)
	r6i.xlarge	emr-5.35.0, emr-6.6.0
	r6i.2xlarge	emr-5.35.0, emr-6.6.0
	r6i.4xlarge	emr-5.35.0, emr-6.6.0
	r6i.8xlarge	emr-5.35.0, emr-6.6.0
	r6i.12xlarge	emr-5.35.0, emr-6.6.0
	r6i.16xlarge	emr-5.35.0, emr-6.6.0
	r6i.24xlarge	emr-5.35.0, emr-6.6.0
	r6i.32xlarge	emr-5.35.0, emr-6.6.0
	r7g.xlarge	emr-5.36.1, emr-6.10.0, emr-7.0.0
	r7g.2xlarge	emr-5.36.1, emr-6.10.0, emr-7.0.0
	r7g.4xlarge	emr-5.36.1, emr-6.10.0, emr-7.0.0
	r7g.8xlarge	emr-5.36.1, emr-6.10.0, emr-7.0.0
	r7g.12xlarge	emr-5.36.1, emr-6.10.0, emr-7.0.0
	r7g.16xlarge	emr-5.36.1, emr-6.10.0, emr-7.0.0
	x1.16xlarge	emr-5.36.1, emr-6.9.0, emr-7.0.0

Kelas Instans	Tipe instans	Versi EMR Amazon yang didukung minimum (5.x, 6.x, 7.x)
	x1.32xlarge	emr-5.36.1, emr-6.9.0, emr-7.0.0
	x2idn.16xlarge	emr-5.36.1, emr-6.9.0
	x2idn.24xlarge	emr-5.36.1, emr-6.9.0
	x2idn.32xlarge	emr-5.36.1, emr-6.9.0
	x2iedn.xlarge	emr-5.36.1, emr-6.9.0
	x2iedn.2xlarge	emr-5.36.1, emr-6.9.0
	x2iedn.4xlarge	emr-5.36.1, emr-6.9.0
	x2iedn.8xlarge	emr-5.36.1, emr-6.9.0
	x2iedn.16xlarge	emr-5.36.1, emr-6.9.0
	x2iedn.24xlarge	emr-5.36.1, emr-6.9.0
	x2iedn.32xlarge	emr-5.36.1, emr-6.9.0
	z1d.xlarge	emr-5.17.0, emr-6.0.0, emr-7.0.0
	z1d.2xlarge	emr-5.17.0, emr-6.0.0, emr-7.0.0
	z1d.3xlarge	emr-5.17.0, emr-6.0.0, emr-7.0.0
	z1d.6xlarge	emr-5.17.0, emr-6.0.0, emr-7.0.0
	z1d.12xlarge	emr-5.17.0, emr-6.0.0, emr-7.0.0

Kelas Instans	Tipe instans	Versi EMR Amazon yang didukung minimum (5.x, 6.x, 7.x)
Penyimpanan Dioptimalkan	i3.xlarge	emr-5.9.0, emr-6.0.0, emr-7.0.0
	i3.2xlarge	emr-5.9.0, emr-6.0.0, emr-7.0.0
	i3.4xlarge	emr-5.9.0, emr-6.0.0, emr-7.0.0
	i3.8xlarge	emr-5.9.0, emr-6.0.0, emr-7.0.0
	i3.16xlarge	emr-5.9.0, emr-6.0.0, emr-7.0.0
	i3en.xlarge	emr-5.25.0, emr-6.0.0, emr-7.0.0
	i3en.2xlarge	emr-5.25.0, emr-6.0.0, emr-7.0.0
	i3en.3xlarge	emr-5.25.0, emr-6.0.0, emr-7.0.0
	i3en.6xlarge	emr-5.25.0, emr-6.0.0, emr-7.0.0
	i3en.12xlarge	emr-5.25.0, emr-6.0.0, emr-7.0.0
	i3en.24xlarge	emr-5.25.0, emr-6.0.0, emr-7.0.0
	i4i.xlarge	emr-5.36.1, emr-6.8.0
	i4i.2xlarge	emr-5.36.1, emr-6.8.0

Kelas Instans	Tipe instans	Versi EMR Amazon yang didukung minimum (5.x, 6.x, 7.x)
	i4i.4xlarge	emr-5.36.1, emr-6.8.0
	i4i.8xlarge	emr-5.36.1, emr-6.8.0
	i4i.12xlarge	emr-5.36.1, emr-6.10.0
	i4i.16xlarge	emr-5.36.1, emr-6.8.0
	i4i.24xlarge	emr-5.36.1, emr-6.10.0
	i4i.32xlarge	emr-5.36.1, emr-6.8.0

Tiongkok (Beijing) - cn-utara-1

Kelas Instans	Tipe instans	Versi EMR Amazon yang didukung minimum (5.x, 6.x, 7.x)
Tujuan Umum	m5.xlarge	emr-5.13.0, emr-6.0.0, emr-7.0.0
	m5.2xlarge	emr-5.13.0, emr-6.0.0, emr-7.0.0
	m5.4xlarge	emr-5.13.0, emr-6.0.0, emr-7.0.0
	m5.8xlarge	emr-5.13.0, emr-6.0.0, emr-7.0.0
	m5.12xlarge	emr-5.13.0, emr-6.0.0, emr-7.0.0
	m5.16xlarge	emr-5.13.0, emr-6.0.0, emr-7.0.0

Kelas Instans	Tipe instans	Versi EMR Amazon yang didukung minimum (5.x, 6.x, 7.x)
	m5.24xlarge	emr-5.13.0, emr-6.0.0, emr-7.0.0
	m5a.xlarge	emr-5.20.0, emr-6.0.0, emr-7.0.0
	m5a.2xlarge	emr-5.20.0, emr-6.0.0, emr-7.0.0
	m5a.4xlarge	emr-5.20.0, emr-6.0.0, emr-7.0.0
	m5a.8xlarge	emr-5.20.0, emr-6.0.0, emr-7.0.0
	m5a.12xlarge	emr-5.20.0, emr-6.0.0, emr-7.0.0
	m5a.16xlarge	emr-5.20.0, emr-6.0.0, emr-7.0.0
	m5a.24xlarge	emr-5.20.0, emr-6.0.0, emr-7.0.0
	m5d.xlarge	emr-5.13.0, emr-6.0.0, emr-7.0.0
	m5d.2xlarge	emr-5.13.0, emr-6.0.0, emr-7.0.0
	m5d.4xlarge	emr-5.13.0, emr-6.0.0, emr-7.0.0
	m5d.8xlarge	emr-5.13.0, emr-6.0.0, emr-7.0.0

Kelas Instans	Tipe instans	Versi EMR Amazon yang didukung minimum (5.x, 6.x, 7.x)
	m5d.12xlarge	emr-5.13.0, emr-6.0.0, emr-7.0.0
	m5d.16xlarge	emr-5.13.0, emr-6.0.0, emr-7.0.0
	m5d.24xlarge	emr-5.13.0, emr-6.0.0, emr-7.0.0
	m6g.xlarge	emr-5.30.0, emr-6.1.0, emr-7.0.0
	m6g.2xlarge	emr-5.30.0, emr-6.1.0, emr-7.0.0
	m6g.4xlarge	emr-5.30.0, emr-6.1.0, emr-7.0.0
	m6g.8xlarge	emr-5.30.0, emr-6.1.0, emr-7.0.0
	m6g.12xlarge	emr-5.30.0, emr-6.1.0, emr-7.0.0
	m6g.16xlarge	emr-5.30.0, emr-6.1.0, emr-7.0.0
	m6i.xlarge	emr-5.35.0, emr-6.6.0
	m6i.2xlarge	emr-5.35.0, emr-6.6.0
	m6i.4xlarge	emr-5.35.0, emr-6.6.0
	m6i.8xlarge	emr-5.35.0, emr-6.6.0
	m6i.12xlarge	emr-5.35.0, emr-6.6.0

Kelas Instans	Tipe instans	Versi EMR Amazon yang didukung minimum (5.x, 6.x, 7.x)
	m6i.16xlarge	emr-5.35.0, emr-6.6.0
	m6i.24xlarge	emr-5.35.0, emr-6.6.0
	m6i.32xlarge	emr-5.35.0, emr-6.6.0
	m7g.xlarge	emr-5.36.1, emr-6.10.0, emr-7.0.0
	m7g.2xlarge	emr-5.36.1, emr-6.10.0, emr-7.0.0
	m7g.4xlarge	emr-5.36.1, emr-6.10.0, emr-7.0.0
	m7g.8xlarge	emr-5.36.1, emr-6.10.0, emr-7.0.0
	m7g.12xlarge	emr-5.36.1, emr-6.10.0, emr-7.0.0
	m7g.16xlarge	emr-5.36.1, emr-6.10.0, emr-7.0.0
Komputasi Dioptimalkan	c5.xlarge	emr-5.13.0, emr-6.0.0, emr-7.0.0
	c5.2xlarge	emr-5.13.0, emr-6.0.0, emr-7.0.0
	c5.4xlarge	emr-5.13.0, emr-6.0.0, emr-7.0.0
	c5.9xlarge	emr-5.13.0, emr-6.0.0, emr-7.0.0

Kelas Instans	Tipe instans	Versi EMR Amazon yang didukung minimum (5.x, 6.x, 7.x)
	c5.12xlarge	emr-5.13.0, emr-6.0.0, emr-7.0.0
	c5.18xlarge	emr-5.13.0, emr-6.0.0, emr-7.0.0
	c5.24xlarge	emr-5.13.0, emr-6.0.0, emr-7.0.0
	c5a.xlarge	emr-5.31.0, emr-6.1.0, emr-7.0.0
	c5a.2xlarge	emr-5.31.0, emr-6.1.0, emr-7.0.0
	c5a.4xlarge	emr-5.31.0, emr-6.1.0, emr-7.0.0
	c5a.8xlarge	emr-5.31.0, emr-6.1.0, emr-7.0.0
	c5a.12xlarge	emr-5.31.0, emr-6.1.0, emr-7.0.0
	c5a.16xlarge	emr-5.31.0, emr-6.1.0, emr-7.0.0
	c5a.24xlarge	emr-5.31.0, emr-6.1.0, emr-7.0.0
	c5d.xlarge	emr-5.13.0, emr-6.0.0, emr-7.0.0
	c5d.2xlarge	emr-5.13.0, emr-6.0.0, emr-7.0.0

Kelas Instans	Tipe instans	Versi EMR Amazon yang didukung minimum (5.x, 6.x, 7.x)
	c5d.4xlarge	emr-5.13.0, emr-6.0.0, emr-7.0.0
	c5d.9xlarge	emr-5.13.0, emr-6.0.0, emr-7.0.0
	c5d.12xlarge	emr-5.13.0, emr-6.0.0, emr-7.0.0
	c5d.18xlarge	emr-5.13.0, emr-6.0.0, emr-7.0.0
	c5d.24xlarge	emr-5.13.0, emr-6.0.0, emr-7.0.0
	c6g.xlarge	emr-5.31.0, emr-6.1.0, emr-7.0.0
	c6g.2xlarge	emr-5.31.0, emr-6.1.0, emr-7.0.0
	c6g.4xlarge	emr-5.31.0, emr-6.1.0, emr-7.0.0
	c6g.8xlarge	emr-5.31.0, emr-6.1.0, emr-7.0.0
	c6g.12xlarge	emr-5.31.0, emr-6.1.0, emr-7.0.0
	c6g.16xlarge	emr-5.31.0, emr-6.1.0, emr-7.0.0
	c6i.xlarge	emr-5.35.0, emr-6.6.0
	c6i.2xlarge	emr-5.35.0, emr-6.6.0

Kelas Instans	Tipe instans	Versi EMR Amazon yang didukung minimum (5.x, 6.x, 7.x)
	c6i.4xlarge	emr-5.35.0, emr-6.6.0
	c6i.8xlarge	emr-5.35.0, emr-6.6.0
	c6i.12xlarge	emr-5.35.0, emr-6.6.0
	c6i.16xlarge	emr-5.35.0, emr-6.6.0
	c6i.24xlarge	emr-5.35.0, emr-6.6.0
	c6i.32xlarge	emr-5.35.0, emr-6.6.0
	c7g.xlarge	emr-5.36.1, emr-6.7.0, emr-7.0.0
	c7g.2xlarge	emr-5.36.1, emr-6.7.0, emr-7.0.0
	c7g.4xlarge	emr-5.36.1, emr-6.7.0, emr-7.0.0
	c7g.8xlarge	emr-5.36.1, emr-6.7.0, emr-7.0.0
	c7g.12xlarge	emr-5.36.1, emr-6.7.0, emr-7.0.0
	c7g.16xlarge	emr-5.36.1, emr-6.7.0, emr-7.0.0
Komputasi yang Dipercepat	g3.4xlarge	emr-5.18.0, emr-6.0.0, emr-7.0.0
	g3.8xlarge	emr-5.18.0, emr-6.0.0, emr-7.0.0

Kelas Instans	Tipe instans	Versi EMR Amazon yang didukung minimum (5.x, 6.x, 7.x)
	g3.16xlarge	emr-5.18.0, emr-6.0.0, emr-7.0.0
	g3s.xlarge	emr-5.19.0, emr-6.0.0, emr-7.0.0
	g4dn.xlarge	emr-5.30.0, emr-6.0.0, emr-7.0.0
	g4dn.2xlarge	emr-5.30.0, emr-6.0.0, emr-7.0.0
	g4dn.4xlarge	emr-5.30.0, emr-6.0.0, emr-7.0.0
	g4dn.8xlarge	emr-5.30.0, emr-6.0.0, emr-7.0.0
	g4dn.12xlarge	emr-5.30.0, emr-6.0.0, emr-7.0.0
	g4dn.16xlarge	emr-5.30.0, emr-6.0.0, emr-7.0.0
	p2.xlarge	emr-5.10.0, emr-6.0.0, emr-7.0.0
	p2.8xlarge	emr-5.10.0, emr-6.0.0, emr-7.0.0
	p2.16xlarge	emr-5.10.0, emr-6.0.0, emr-7.0.0
	p3.2xlarge	emr-5.10.0, emr-6.0.0, emr-7.0.0

Kelas Instans	Tipe instans	Versi EMR Amazon yang didukung minimum (5.x, 6.x, 7.x)
	p3.8xlarge	emr-5.10.0, emr-6.0.0, emr-7.0.0
	p3.16xlarge	emr-5.10.0, emr-6.0.0, emr-7.0.0
Memori Dioptimalkan	r5.xlarge	emr-5.13.0, emr-6.0.0, emr-7.0.0
	r5.2xlarge	emr-5.13.0, emr-6.0.0, emr-7.0.0
	r5.4xlarge	emr-5.13.0, emr-6.0.0, emr-7.0.0
	r5.8xlarge	emr-5.13.0, emr-6.0.0, emr-7.0.0
	r5.12xlarge	emr-5.13.0, emr-6.0.0, emr-7.0.0
	r5.16xlarge	emr-5.13.0, emr-6.0.0, emr-7.0.0
	r5.24xlarge	emr-5.13.0, emr-6.0.0, emr-7.0.0
	r5a.xlarge	emr-5.20.0, emr-6.0.0, emr-7.0.0
	r5a.2xlarge	emr-5.20.0, emr-6.0.0, emr-7.0.0
	r5a.4xlarge	emr-5.20.0, emr-6.0.0, emr-7.0.0

Kelas Instans	Tipe instans	Versi EMR Amazon yang didukung minimum (5.x, 6.x, 7.x)
	r5a.8xlarge	emr-5.20.0, emr-6.0.0, emr-7.0.0
	r5a.12xlarge	emr-5.20.0, emr-6.0.0, emr-7.0.0
	r5a.16xlarge	emr-5.20.0, emr-6.0.0, emr-7.0.0
	r5a.24xlarge	emr-5.20.0, emr-6.0.0, emr-7.0.0
	r5d.xlarge	emr-5.13.0, emr-6.0.0, emr-7.0.0
	r5d.2xlarge	emr-5.13.0, emr-6.0.0, emr-7.0.0
	r5d.4xlarge	emr-5.13.0, emr-6.0.0, emr-7.0.0
	r5d.8xlarge	emr-5.13.0, emr-6.0.0, emr-7.0.0
	r5d.12xlarge	emr-5.13.0, emr-6.0.0, emr-7.0.0
	r5d.16xlarge	emr-5.13.0, emr-6.0.0, emr-7.0.0
	r5d.24xlarge	emr-5.13.0, emr-6.0.0, emr-7.0.0
	r6g.xlarge	emr-5.31.0, emr-6.1.0, emr-7.0.0

Kelas Instans	Tipe instans	Versi EMR Amazon yang didukung minimum (5.x, 6.x, 7.x)
	r6g.2xlarge	emr-5.31.0, emr-6.1.0, emr-7.0.0
	r6g.4xlarge	emr-5.31.0, emr-6.1.0, emr-7.0.0
	r6g.8xlarge	emr-5.31.0, emr-6.1.0, emr-7.0.0
	r6g.12xlarge	emr-5.31.0, emr-6.1.0, emr-7.0.0
	r6g.16xlarge	emr-5.31.0, emr-6.1.0, emr-7.0.0
	r6gd.xlarge	emr-5.33.0, emr-6.3.0, emr-7.0.0
	r6gd.2xlarge	emr-5.33.0, emr-6.3.0, emr-7.0.0
	r6gd.4xlarge	emr-5.33.0, emr-6.3.0, emr-7.0.0
	r6gd.8xlarge	emr-5.33.0, emr-6.3.0, emr-7.0.0
	r6gd.12xlarge	emr-5.33.0, emr-6.3.0, emr-7.0.0
	r6gd.16xlarge	emr-5.33.0, emr-6.3.0, emr-7.0.0
	r6i.xlarge	emr-5.35.0, emr-6.6.0
	r6i.2xlarge	emr-5.35.0, emr-6.6.0

Kelas Instans	Tipe instans	Versi EMR Amazon yang didukung minimum (5.x, 6.x, 7.x)
	r6i.4xlarge	emr-5.35.0, emr-6.6.0
	r6i.8xlarge	emr-5.35.0, emr-6.6.0
	r6i.12xlarge	emr-5.35.0, emr-6.6.0
	r6i.16xlarge	emr-5.35.0, emr-6.6.0
	r6i.24xlarge	emr-5.35.0, emr-6.6.0
	r6i.32xlarge	emr-5.35.0, emr-6.6.0
	r7g.xlarge	emr-5.36.1, emr-6.10.0, emr-7.0.0
	r7g.2xlarge	emr-5.36.1, emr-6.10.0, emr-7.0.0
	r7g.4xlarge	emr-5.36.1, emr-6.10.0, emr-7.0.0
	r7g.8xlarge	emr-5.36.1, emr-6.10.0, emr-7.0.0
	r7g.12xlarge	emr-5.36.1, emr-6.10.0, emr-7.0.0
	r7g.16xlarge	emr-5.36.1, emr-6.10.0, emr-7.0.0
	x1.16xlarge	emr-5.36.1, emr-6.9.0, emr-7.0.0
	x1.32xlarge	emr-5.36.1, emr-6.9.0, emr-7.0.0

Kelas Instans	Tipe instans	Versi EMR Amazon yang didukung minimum (5.x, 6.x, 7.x)
	x2idn.16xlarge	emr-5.36.1, emr-6.9.0
	x2idn.24xlarge	emr-5.36.1, emr-6.9.0
	x2idn.32xlarge	emr-5.36.1, emr-6.9.0
	x2iedn.xlarge	emr-5.36.1, emr-6.9.0
	x2iedn.2xlarge	emr-5.36.1, emr-6.9.0
	x2iedn.4xlarge	emr-5.36.1, emr-6.9.0
	x2iedn.8xlarge	emr-5.36.1, emr-6.9.0
	x2iedn.16xlarge	emr-5.36.1, emr-6.9.0
	x2iedn.24xlarge	emr-5.36.1, emr-6.9.0
	x2iedn.32xlarge	emr-5.36.1, emr-6.9.0
Penyimpanan Dioptimalkan	i3.xlarge	emr-5.9.0, emr-6.0.0, emr-7.0.0
	i3.2xlarge	emr-5.9.0, emr-6.0.0, emr-7.0.0
	i3.4xlarge	emr-5.9.0, emr-6.0.0, emr-7.0.0
	i3.8xlarge	emr-5.9.0, emr-6.0.0, emr-7.0.0
	i3.16xlarge	emr-5.9.0, emr-6.0.0, emr-7.0.0
	i3en.xlarge	emr-5.25.0, emr-6.0.0, emr-7.0.0

Kelas Instans	Tipe instans	Versi EMR Amazon yang didukung minimum (5.x, 6.x, 7.x)
	i3en.2xlarge	emr-5.25.0, emr-6.0.0, emr-7.0.0
	i3en.3xlarge	emr-5.25.0, emr-6.0.0, emr-7.0.0
	i3en.6xlarge	emr-5.25.0, emr-6.0.0, emr-7.0.0
	i3en.12xlarge	emr-5.25.0, emr-6.0.0, emr-7.0.0
	i3en.24xlarge	emr-5.25.0, emr-6.0.0, emr-7.0.0
	i4i.xlarge	emr-5.36.1, emr-6.8.0
	i4i.2xlarge	emr-5.36.1, emr-6.8.0
	i4i.4xlarge	emr-5.36.1, emr-6.8.0
	i4i.8xlarge	emr-5.36.1, emr-6.8.0
	i4i.12xlarge	emr-5.36.1, emr-6.10.0
	i4i.16xlarge	emr-5.36.1, emr-6.8.0
	i4i.24xlarge	emr-5.36.1, emr-6.10.0
	i4i.32xlarge	emr-5.36.1, emr-6.8.0

Europe (Frankfurt) - eu-central-1

Kelas Instans	Tipe instans	Versi EMR Amazon yang didukung minimum (5.x, 6.x, 7.x)
Tujuan Umum	m5.xlarge	emr-5.13.0, emr-6.0.0, emr-7.0.0
	m5.2xlarge	emr-5.13.0, emr-6.0.0, emr-7.0.0
	m5.4xlarge	emr-5.13.0, emr-6.0.0, emr-7.0.0
	m5.8xlarge	emr-5.13.0, emr-6.0.0, emr-7.0.0
	m5.12xlarge	emr-5.13.0, emr-6.0.0, emr-7.0.0
	m5.16xlarge	emr-5.13.0, emr-6.0.0, emr-7.0.0
	m5.24xlarge	emr-5.13.0, emr-6.0.0, emr-7.0.0
	m5a.xlarge	emr-5.20.0, emr-6.0.0, emr-7.0.0
	m5a.2xlarge	emr-5.20.0, emr-6.0.0, emr-7.0.0
	m5a.4xlarge	emr-5.20.0, emr-6.0.0, emr-7.0.0
	m5a.8xlarge	emr-5.20.0, emr-6.0.0, emr-7.0.0

Kelas Instans	Tipe instans	Versi EMR Amazon yang didukung minimum (5.x, 6.x, 7.x)
	m5a.12xlarge	emr-5.20.0, emr-6.0.0, emr-7.0.0
	m5a.16xlarge	emr-5.20.0, emr-6.0.0, emr-7.0.0
	m5a.24xlarge	emr-5.20.0, emr-6.0.0, emr-7.0.0
	m5ad.xlarge	emr-5.20.0, emr-6.0.0, emr-7.0.0
	m5ad.2xlarge	emr-5.20.0, emr-6.0.0, emr-7.0.0
	m5ad.4xlarge	emr-5.20.0, emr-6.0.0, emr-7.0.0
	m5ad.8xlarge	emr-5.33.0, emr-6.3.0, emr-7.0.0
	m5ad.12xlarge	emr-5.20.0, emr-6.0.0, emr-7.0.0
	m5ad.16xlarge	emr-5.33.0, emr-6.3.0, emr-7.0.0
	m5ad.24xlarge	emr-5.20.0, emr-6.0.0, emr-7.0.0
	m5d.xlarge	emr-5.13.0, emr-6.0.0, emr-7.0.0
	m5d.2xlarge	emr-5.13.0, emr-6.0.0, emr-7.0.0

Kelas Instans	Tipe instans	Versi EMR Amazon yang didukung minimum (5.x, 6.x, 7.x)
	m5d.4xlarge	emr-5.13.0, emr-6.0.0, emr-7.0.0
	m5d.8xlarge	emr-5.13.0, emr-6.0.0, emr-7.0.0
	m5d.12xlarge	emr-5.13.0, emr-6.0.0, emr-7.0.0
	m5d.16xlarge	emr-5.13.0, emr-6.0.0, emr-7.0.0
	m5d.24xlarge	emr-5.13.0, emr-6.0.0, emr-7.0.0
	m5dn.xlarge	emr-5.31.0, emr-6.1.0, emr-7.0.0
	m5dn.2xlarge	emr-5.31.0, emr-6.1.0, emr-7.0.0
	m5dn.4xlarge	emr-5.31.0, emr-6.1.0, emr-7.0.0
	m5dn.8xlarge	emr-5.31.0, emr-6.1.0, emr-7.0.0
	m5dn.12xlarge	emr-5.31.0, emr-6.1.0, emr-7.0.0
	m5dn.16xlarge	emr-5.31.0, emr-6.1.0, emr-7.0.0
	m5dn.24xlarge	emr-5.31.0, emr-6.1.0, emr-7.0.0

Kelas Instans	Tipe instans	Versi EMR Amazon yang didukung minimum (5.x, 6.x, 7.x)
	m5n.xlarge	emr-5.31.0, emr-6.1.0, emr-7.0.0
	m5n.2xlarge	emr-5.31.0, emr-6.1.0, emr-7.0.0
	m5n.4xlarge	emr-5.31.0, emr-6.1.0, emr-7.0.0
	m5n.8xlarge	emr-5.31.0, emr-6.1.0, emr-7.0.0
	m5n.12xlarge	emr-5.31.0, emr-6.1.0, emr-7.0.0
	m5n.16xlarge	emr-5.31.0, emr-6.1.0, emr-7.0.0
	m5n.24xlarge	emr-5.31.0, emr-6.1.0, emr-7.0.0
	m5zn.xlarge	emr-5.33.0, emr-6.3.0, emr-7.0.0
	m5zn.2xlarge	emr-5.33.0, emr-6.3.0, emr-7.0.0
	m5zn.3xlarge	emr-5.33.0, emr-6.3.0, emr-7.0.0
	m5zn.6xlarge	emr-5.33.0, emr-6.3.0, emr-7.0.0
	m5zn.12xlarge	emr-5.33.0, emr-6.3.0, emr-7.0.0

Kelas Instans	Tipe instans	Versi EMR Amazon yang didukung minimum (5.x, 6.x, 7.x)
	m6a.xlarge	emr-5.36.1, emr-6.8.0, emr-7.0.0
	m6a.2xlarge	emr-5.36.1, emr-6.8.0, emr-7.0.0
	m6a.4xlarge	emr-5.36.1, emr-6.8.0, emr-7.0.0
	m6a.8xlarge	emr-5.36.1, emr-6.8.0, emr-7.0.0
	m6a.12xlarge	emr-5.36.1, emr-6.8.0, emr-7.0.0
	m6a.16xlarge	emr-5.36.1, emr-6.8.0, emr-7.0.0
	m6a.24xlarge	emr-5.36.1, emr-6.8.0, emr-7.0.0
	m6a.32xlarge	emr-5.36.1, emr-6.8.0, emr-7.0.0
	m6a.48xlarge	emr-5.36.1, emr-6.8.0, emr-7.0.0
	m6g.xlarge	emr-5.30.0, emr-6.1.0, emr-7.0.0
	m6g.2xlarge	emr-5.30.0, emr-6.1.0, emr-7.0.0
	m6g.4xlarge	emr-5.30.0, emr-6.1.0, emr-7.0.0

Kelas Instans	Tipe instans	Versi EMR Amazon yang didukung minimum (5.x, 6.x, 7.x)
	m6g.8xlarge	emr-5.30.0, emr-6.1.0, emr-7.0.0
	m6g.12xlarge	emr-5.30.0, emr-6.1.0, emr-7.0.0
	m6g.16xlarge	emr-5.30.0, emr-6.1.0, emr-7.0.0
	m6gd.xlarge	emr-5.33.0, emr-6.3.0, emr-7.0.0
	m6gd.2xlarge	emr-5.33.0, emr-6.3.0, emr-7.0.0
	m6gd.4xlarge	emr-5.33.0, emr-6.3.0, emr-7.0.0
	m6gd.8xlarge	emr-5.33.0, emr-6.3.0, emr-7.0.0
	m6gd.12xlarge	emr-5.33.0, emr-6.3.0, emr-7.0.0
	m6gd.16xlarge	emr-5.33.0, emr-6.3.0, emr-7.0.0
	m6i.xlarge	emr-5.35.0, emr-6.6.0
	m6i.2xlarge	emr-5.35.0, emr-6.6.0
	m6i.4xlarge	emr-5.35.0, emr-6.6.0
	m6i.8xlarge	emr-5.35.0, emr-6.6.0
	m6i.12xlarge	emr-5.35.0, emr-6.6.0

Kelas Instans	Tipe instans	Versi EMR Amazon yang didukung minimum (5.x, 6.x, 7.x)
	m6i.16xlarge	emr-5.35.0, emr-6.6.0
	m6i.24xlarge	emr-5.35.0, emr-6.6.0
	m6i.32xlarge	emr-5.35.0, emr-6.6.0
	m6id.xlarge	emr-5.36.1, emr-6.8.0
	m6id.2xlarge	emr-5.36.1, emr-6.8.0
	m6id.4xlarge	emr-5.36.1, emr-6.8.0
	m6id.8xlarge	emr-5.36.1, emr-6.8.0
	m6id.12xlarge	emr-5.36.1, emr-6.8.0
	m6id.16xlarge	emr-5.36.1, emr-6.8.0
	m6id.24xlarge	emr-5.36.1, emr-6.8.0
	m6id.32xlarge	emr-5.36.1, emr-6.8.0
	m6idn.xlarge	emr-5.36.1, emr-6.10.0
	m6idn.2xlarge	emr-5.36.1, emr-6.10.0
	m6idn.4xlarge	emr-5.36.1, emr-6.10.0
	m6idn.8xlarge	emr-5.36.1, emr-6.10.0
	m6idn.12xlarge	emr-5.36.1, emr-6.10.0
	m6idn.16xlarge	emr-5.36.1, emr-6.10.0
	m6idn.24xlarge	emr-5.36.1, emr-6.10.0
	m6idn.32xlarge	emr-5.36.1, emr-6.10.0

Kelas Instans	Tipe instans	Versi EMR Amazon yang didukung minimum (5.x, 6.x, 7.x)
	m6in.xlarge	emr-5.36.1, emr-6.10.0
	m6in.2xlarge	emr-5.36.1, emr-6.10.0
	m6in.4xlarge	emr-5.36.1, emr-6.10.0
	m6in.8xlarge	emr-5.36.1, emr-6.10.0
	m6in.12xlarge	emr-5.36.1, emr-6.10.0
	m6in.16xlarge	emr-5.36.1, emr-6.10.0
	m6in.24xlarge	emr-5.36.1, emr-6.10.0
	m6in.32xlarge	emr-5.36.1, emr-6.10.0
	m7g.xlarge	emr-5.36.1, emr-6.10.0, emr-7.0.0
	m7g.2xlarge	emr-5.36.1, emr-6.10.0, emr-7.0.0
	m7g.4xlarge	emr-5.36.1, emr-6.10.0, emr-7.0.0
	m7g.8xlarge	emr-5.36.1, emr-6.10.0, emr-7.0.0
	m7g.12xlarge	emr-5.36.1, emr-6.10.0, emr-7.0.0
	m7g.16xlarge	emr-5.36.1, emr-6.10.0, emr-7.0.0
	m7gd.xlarge	emr-5.36.1, emr-6.10.0, emr-7.0.0

Kelas Instans	Tipe instans	Versi EMR Amazon yang didukung minimum (5.x, 6.x, 7.x)
	m7gd.2xlarge	emr-5.36.1, emr-6.10.0, emr-7.0.0
	m7gd.4xlarge	emr-5.36.1, emr-6.10.0, emr-7.0.0
	m7gd.8xlarge	emr-5.36.1, emr-6.10.0, emr-7.0.0
	m7gd.12xlarge	emr-5.36.1, emr-6.10.0, emr-7.0.0
	m7gd.16xlarge	emr-5.36.1, emr-6.10.0, emr-7.0.0
Komputasi Dioptimalkan	c5.xlarge	emr-5.13.0, emr-6.0.0, emr-7.0.0
	c5.2xlarge	emr-5.13.0, emr-6.0.0, emr-7.0.0
	c5.4xlarge	emr-5.13.0, emr-6.0.0, emr-7.0.0
	c5.9xlarge	emr-5.13.0, emr-6.0.0, emr-7.0.0
	c5.12xlarge	emr-5.13.0, emr-6.0.0, emr-7.0.0
	c5.18xlarge	emr-5.13.0, emr-6.0.0, emr-7.0.0
	c5.24xlarge	emr-5.13.0, emr-6.0.0, emr-7.0.0

Kelas Instans	Tipe instans	Versi EMR Amazon yang didukung minimum (5.x, 6.x, 7.x)
	c5a.xlarge	emr-5.31.0, emr-6.1.0, emr-7.0.0
	c5a.2xlarge	emr-5.31.0, emr-6.1.0, emr-7.0.0
	c5a.4xlarge	emr-5.31.0, emr-6.1.0, emr-7.0.0
	c5a.8xlarge	emr-5.31.0, emr-6.1.0, emr-7.0.0
	c5a.12xlarge	emr-5.31.0, emr-6.1.0, emr-7.0.0
	c5a.16xlarge	emr-5.31.0, emr-6.1.0, emr-7.0.0
	c5a.24xlarge	emr-5.31.0, emr-6.1.0, emr-7.0.0
	c5ad.xlarge	emr-5.33.0, emr-6.3.0, emr-7.0.0
	c5ad.2xlarge	emr-5.33.0, emr-6.3.0, emr-7.0.0
	c5ad.4xlarge	emr-5.33.0, emr-6.3.0, emr-7.0.0
	c5ad.8xlarge	emr-5.33.0, emr-6.3.0, emr-7.0.0
	c5ad.12xlarge	emr-5.33.0, emr-6.3.0, emr-7.0.0

Kelas Instans	Tipe instans	Versi EMR Amazon yang didukung minimum (5.x, 6.x, 7.x)
	c5ad.16xlarge	emr-5.33.0, emr-6.3.0, emr-7.0.0
	c5ad.24xlarge	emr-5.33.0, emr-6.3.0, emr-7.0.0
	c5d.xlarge	emr-5.13.0, emr-6.0.0, emr-7.0.0
	c5d.2xlarge	emr-5.13.0, emr-6.0.0, emr-7.0.0
	c5d.4xlarge	emr-5.13.0, emr-6.0.0, emr-7.0.0
	c5d.9xlarge	emr-5.13.0, emr-6.0.0, emr-7.0.0
	c5d.12xlarge	emr-5.13.0, emr-6.0.0, emr-7.0.0
	c5d.18xlarge	emr-5.13.0, emr-6.0.0, emr-7.0.0
	c5d.24xlarge	emr-5.13.0, emr-6.0.0, emr-7.0.0
	c5n.xlarge	emr-5.20.0, emr-6.0.0, emr-7.0.0
	c5n.2xlarge	emr-5.20.0, emr-6.0.0, emr-7.0.0
	c5n.4xlarge	emr-5.20.0, emr-6.0.0, emr-7.0.0

Kelas Instans	Tipe instans	Versi EMR Amazon yang didukung minimum (5.x, 6.x, 7.x)
	c5n.9xlarge	emr-5.20.0, emr-6.0.0, emr-7.0.0
	c5n.18xlarge	emr-5.20.0, emr-6.0.0, emr-7.0.0
	c6a.xlarge	emr-5.36.1, emr-6.8.0, emr-7.0.0
	c6a.2xlarge	emr-5.36.1, emr-6.8.0, emr-7.0.0
	c6a.4xlarge	emr-5.36.1, emr-6.8.0, emr-7.0.0
	c6a.8xlarge	emr-5.36.1, emr-6.8.0, emr-7.0.0
	c6a.12xlarge	emr-5.36.1, emr-6.8.0, emr-7.0.0
	c6a.16xlarge	emr-5.36.1, emr-6.8.0, emr-7.0.0
	c6a.24xlarge	emr-5.36.1, emr-6.8.0, emr-7.0.0
	c6a.32xlarge	emr-5.36.1, emr-6.8.0, emr-7.0.0
	c6a.48xlarge	emr-5.36.1, emr-6.8.0, emr-7.0.0
	c6g.xlarge	emr-5.31.0, emr-6.1.0, emr-7.0.0

Kelas Instans	Tipe instans	Versi EMR Amazon yang didukung minimum (5.x, 6.x, 7.x)
	c6g.2xlarge	emr-5.31.0, emr-6.1.0, emr-7.0.0
	c6g.4xlarge	emr-5.31.0, emr-6.1.0, emr-7.0.0
	c6g.8xlarge	emr-5.31.0, emr-6.1.0, emr-7.0.0
	c6g.12xlarge	emr-5.31.0, emr-6.1.0, emr-7.0.0
	c6g.16xlarge	emr-5.31.0, emr-6.1.0, emr-7.0.0
	c6gd.xlarge	emr-5.33.0, emr-6.3.0, emr-7.0.0
	c6gd.2xlarge	emr-5.33.0, emr-6.3.0, emr-7.0.0
	c6gd.4xlarge	emr-5.33.0, emr-6.3.0, emr-7.0.0
	c6gd.8xlarge	emr-5.33.0, emr-6.3.0, emr-7.0.0
	c6gd.12xlarge	emr-5.33.0, emr-6.3.0, emr-7.0.0
	c6gd.16xlarge	emr-5.33.0, emr-6.3.0, emr-7.0.0
	c6gn.xlarge	emr-5.33.0, emr-6.3.0, emr-7.0.0

Kelas Instans	Tipe instans	Versi EMR Amazon yang didukung minimum (5.x, 6.x, 7.x)
	c6gn.2xlarge	emr-5.33.0, emr-6.3.0, emr-7.0.0
	c6gn.4xlarge	emr-5.33.0, emr-6.3.0, emr-7.0.0
	c6gn.8xlarge	emr-5.33.0, emr-6.3.0, emr-7.0.0
	c6gn.12xlarge	emr-5.33.0, emr-6.3.0, emr-7.0.0
	c6gn.16xlarge	emr-5.33.0, emr-6.3.0, emr-7.0.0
	c6i.xlarge	emr-5.35.0, emr-6.6.0
	c6i.2xlarge	emr-5.35.0, emr-6.6.0
	c6i.4xlarge	emr-5.35.0, emr-6.6.0
	c6i.8xlarge	emr-5.35.0, emr-6.6.0
	c6i.12xlarge	emr-5.35.0, emr-6.6.0
	c6i.16xlarge	emr-5.35.0, emr-6.6.0
	c6i.24xlarge	emr-5.35.0, emr-6.6.0
	c6i.32xlarge	emr-5.35.0, emr-6.6.0
	c6id.xlarge	emr-5.36.1, emr-6.8.0
	c6id.2xlarge	emr-5.36.1, emr-6.8.0
	c6id.4xlarge	emr-5.36.1, emr-6.8.0

Kelas Instans	Tipe instans	Versi EMR Amazon yang didukung minimum (5.x, 6.x, 7.x)
	c6id.8xlarge	emr-5.36.1, emr-6.8.0
	c6id.12xlarge	emr-5.36.1, emr-6.8.0
	c6id.16xlarge	emr-5.36.1, emr-6.8.0
	c6id.24xlarge	emr-5.36.1, emr-6.8.0
	c6id.32xlarge	emr-5.36.1, emr-6.8.0
	c6in.xlarge	emr-5.36.1, emr-6.10.0
	c6in.2xlarge	emr-5.36.1, emr-6.10.0
	c6in.4xlarge	emr-5.36.1, emr-6.10.0
	c6in.8xlarge	emr-5.36.1, emr-6.10.0
	c6in.12xlarge	emr-5.36.1, emr-6.10.0
	c6in.16xlarge	emr-5.36.1, emr-6.10.0
	c6in.24xlarge	emr-5.36.1, emr-6.10.0
	c6in.32xlarge	emr-5.36.1, emr-6.10.0
	c7g.xlarge	emr-5.36.1, emr-6.7.0, emr-7.0.0
	c7g.2xlarge	emr-5.36.1, emr-6.7.0, emr-7.0.0
	c7g.4xlarge	emr-5.36.1, emr-6.7.0, emr-7.0.0
	c7g.8xlarge	emr-5.36.1, emr-6.7.0, emr-7.0.0

Kelas Instans	Tipe instans	Versi EMR Amazon yang didukung minimum (5.x, 6.x, 7.x)
	c7g.12xlarge	emr-5.36.1, emr-6.7.0, emr-7.0.0
	c7g.16xlarge	emr-5.36.1, emr-6.7.0, emr-7.0.0
	c7gd.xlarge	emr-5.36.1, emr-6.10.0, emr-7.0.0
	c7gd.2xlarge	emr-5.36.1, emr-6.10.0, emr-7.0.0
	c7gd.4xlarge	emr-5.36.1, emr-6.10.0, emr-7.0.0
	c7gd.8xlarge	emr-5.36.1, emr-6.10.0, emr-7.0.0
	c7gd.12xlarge	emr-5.36.1, emr-6.10.0, emr-7.0.0
	c7gd.16xlarge	emr-5.36.1, emr-6.10.0, emr-7.0.0
Komputasi yang Dipercepat	g3.4xlarge	emr-5.18.0, emr-6.0.0, emr-7.0.0
	g3.8xlarge	emr-5.18.0, emr-6.0.0, emr-7.0.0
	g3.16xlarge	emr-5.18.0, emr-6.0.0, emr-7.0.0
	g3s.xlarge	emr-5.19.0, emr-6.0.0, emr-7.0.0

Kelas Instans	Tipe instans	Versi EMR Amazon yang didukung minimum (5.x, 6.x, 7.x)
	g4dn.xlarge	emr-5.30.0, emr-6.0.0, emr-7.0.0
	g4dn.2xlarge	emr-5.30.0, emr-6.0.0, emr-7.0.0
	g4dn.4xlarge	emr-5.30.0, emr-6.0.0, emr-7.0.0
	g4dn.8xlarge	emr-5.30.0, emr-6.0.0, emr-7.0.0
	g4dn.12xlarge	emr-5.30.0, emr-6.0.0, emr-7.0.0
	g4dn.16xlarge	emr-5.30.0, emr-6.0.0, emr-7.0.0
	g5.xlarge	emr-5.36.1, emr-6.9.0, emr-7.0.0
	g5.2xlarge	emr-5.36.1, emr-6.9.0, emr-7.0.0
	g5.4xlarge	emr-5.36.1, emr-6.9.0, emr-7.0.0
	g5.8xlarge	emr-5.36.1, emr-6.9.0, emr-7.0.0
	g5.12xlarge	emr-5.36.1, emr-6.9.0, emr-7.0.0
	g5.16xlarge	emr-5.36.1, emr-6.9.0, emr-7.0.0

Kelas Instans	Tipe instans	Versi EMR Amazon yang didukung minimum (5.x, 6.x, 7.x)
	g5.24xlarge	emr-5.36.1, emr-6.9.0, emr-7.0.0
	g5.48xlarge	emr-5.36.1, emr-6.9.0, emr-7.0.0
	p2.xlarge	emr-5.10.0, emr-6.0.0, emr-7.0.0
	p2.8xlarge	emr-5.10.0, emr-6.0.0, emr-7.0.0
	p2.16xlarge	emr-5.10.0, emr-6.0.0, emr-7.0.0
	p3.2xlarge	emr-5.10.0, emr-6.0.0, emr-7.0.0
	p3.8xlarge	emr-5.10.0, emr-6.0.0, emr-7.0.0
	p3.16xlarge	emr-5.10.0, emr-6.0.0, emr-7.0.0
Memori Dioptimalkan	r5.xlarge	emr-5.13.0, emr-6.0.0, emr-7.0.0
	r5.2xlarge	emr-5.13.0, emr-6.0.0, emr-7.0.0
	r5.4xlarge	emr-5.13.0, emr-6.0.0, emr-7.0.0
	r5.8xlarge	emr-5.13.0, emr-6.0.0, emr-7.0.0

Kelas Instans	Tipe instans	Versi EMR Amazon yang didukung minimum (5.x, 6.x, 7.x)
	r5.12xlarge	emr-5.13.0, emr-6.0.0, emr-7.0.0
	r5.16xlarge	emr-5.13.0, emr-6.0.0, emr-7.0.0
	r5.24xlarge	emr-5.13.0, emr-6.0.0, emr-7.0.0
	r5a.xlarge	emr-5.20.0, emr-6.0.0, emr-7.0.0
	r5a.2xlarge	emr-5.20.0, emr-6.0.0, emr-7.0.0
	r5a.4xlarge	emr-5.20.0, emr-6.0.0, emr-7.0.0
	r5a.8xlarge	emr-5.20.0, emr-6.0.0, emr-7.0.0
	r5a.12xlarge	emr-5.20.0, emr-6.0.0, emr-7.0.0
	r5a.16xlarge	emr-5.20.0, emr-6.0.0, emr-7.0.0
	r5a.24xlarge	emr-5.20.0, emr-6.0.0, emr-7.0.0
	r5ad.xlarge	emr-5.20.0, emr-6.0.0, emr-7.0.0
	r5ad.2xlarge	emr-5.20.0, emr-6.0.0, emr-7.0.0

Kelas Instans	Tipe instans	Versi EMR Amazon yang didukung minimum (5.x, 6.x, 7.x)
	r5ad.4xlarge	emr-5.20.0, emr-6.0.0, emr-7.0.0
	r5ad.8xlarge	emr-5.33.0, emr-6.3.0, emr-7.0.0
	r5ad.12xlarge	emr-5.20.0, emr-6.0.0, emr-7.0.0
	r5ad.16xlarge	emr-5.33.0, emr-6.3.0, emr-7.0.0
	r5ad.24xlarge	emr-5.20.0, emr-6.0.0, emr-7.0.0
	r5b.xlarge	emr-5.33.0, emr-6.3.0, emr-7.0.0
	r5b.2xlarge	emr-5.33.0, emr-6.3.0, emr-7.0.0
	r5b.4xlarge	emr-5.33.0, emr-6.3.0, emr-7.0.0
	r5b.8xlarge	emr-5.33.0, emr-6.3.0, emr-7.0.0
	r5b.12xlarge	emr-5.33.0, emr-6.3.0, emr-7.0.0
	r5b.16xlarge	emr-5.33.0, emr-6.3.0, emr-7.0.0
	r5b.24xlarge	emr-5.33.0, emr-6.3.0, emr-7.0.0

Kelas Instans	Tipe instans	Versi EMR Amazon yang didukung minimum (5.x, 6.x, 7.x)
	r5d.xlarge	emr-5.13.0, emr-6.0.0, emr-7.0.0
	r5d.2xlarge	emr-5.13.0, emr-6.0.0, emr-7.0.0
	r5d.4xlarge	emr-5.13.0, emr-6.0.0, emr-7.0.0
	r5d.8xlarge	emr-5.13.0, emr-6.0.0, emr-7.0.0
	r5d.12xlarge	emr-5.13.0, emr-6.0.0, emr-7.0.0
	r5d.16xlarge	emr-5.13.0, emr-6.0.0, emr-7.0.0
	r5d.24xlarge	emr-5.13.0, emr-6.0.0, emr-7.0.0
	r5dn.xlarge	emr-5.31.0, emr-6.1.0, emr-7.0.0
	r5dn.2xlarge	emr-5.31.0, emr-6.1.0, emr-7.0.0
	r5dn.4xlarge	emr-5.31.0, emr-6.1.0, emr-7.0.0
	r5dn.8xlarge	emr-5.31.0, emr-6.1.0, emr-7.0.0
	r5dn.12xlarge	emr-5.31.0, emr-6.1.0, emr-7.0.0

Kelas Instans	Tipe instans	Versi EMR Amazon yang didukung minimum (5.x, 6.x, 7.x)
	r5dn.16xlarge	emr-5.31.0, emr-6.1.0, emr-7.0.0
	r5dn.24xlarge	emr-5.31.0, emr-6.1.0, emr-7.0.0
	r5n.xlarge	emr-5.31.0, emr-6.1.0, emr-7.0.0
	r5n.2xlarge	emr-5.31.0, emr-6.1.0, emr-7.0.0
	r5n.4xlarge	emr-5.31.0, emr-6.1.0, emr-7.0.0
	r5n.8xlarge	emr-5.31.0, emr-6.1.0, emr-7.0.0
	r5n.12xlarge	emr-5.31.0, emr-6.1.0, emr-7.0.0
	r5n.16xlarge	emr-5.31.0, emr-6.1.0, emr-7.0.0
	r5n.24xlarge	emr-5.31.0, emr-6.1.0, emr-7.0.0
	r6a.xlarge	emr-5.36.1, emr-6.8.0, emr-7.0.0
	r6a.2xlarge	emr-5.36.1, emr-6.8.0, emr-7.0.0
	r6a.4xlarge	emr-5.36.1, emr-6.8.0, emr-7.0.0

Kelas Instans	Tipe instans	Versi EMR Amazon yang didukung minimum (5.x, 6.x, 7.x)
	r6a.8xlarge	emr-5.36.1, emr-6.8.0, emr-7.0.0
	r6a.12xlarge	emr-5.36.1, emr-6.8.0, emr-7.0.0
	r6a.16xlarge	emr-5.36.1, emr-6.8.0, emr-7.0.0
	r6a.24xlarge	emr-5.36.1, emr-6.8.0, emr-7.0.0
	r6a.32xlarge	emr-5.36.1, emr-6.8.0, emr-7.0.0
	r6a.48xlarge	emr-5.36.1, emr-6.8.0, emr-7.0.0
	r6g.xlarge	emr-5.31.0, emr-6.1.0, emr-7.0.0
	r6g.2xlarge	emr-5.31.0, emr-6.1.0, emr-7.0.0
	r6g.4xlarge	emr-5.31.0, emr-6.1.0, emr-7.0.0
	r6g.8xlarge	emr-5.31.0, emr-6.1.0, emr-7.0.0
	r6g.12xlarge	emr-5.31.0, emr-6.1.0, emr-7.0.0
	r6g.16xlarge	emr-5.31.0, emr-6.1.0, emr-7.0.0

Kelas Instans	Tipe instans	Versi EMR Amazon yang didukung minimum (5.x, 6.x, 7.x)
	r6gd.xlarge	emr-5.33.0, emr-6.3.0, emr-7.0.0
	r6gd.2xlarge	emr-5.33.0, emr-6.3.0, emr-7.0.0
	r6gd.4xlarge	emr-5.33.0, emr-6.3.0, emr-7.0.0
	r6gd.8xlarge	emr-5.33.0, emr-6.3.0, emr-7.0.0
	r6gd.12xlarge	emr-5.33.0, emr-6.3.0, emr-7.0.0
	r6gd.16xlarge	emr-5.33.0, emr-6.3.0, emr-7.0.0
	r6i.xlarge	emr-5.35.0, emr-6.6.0
	r6i.2xlarge	emr-5.35.0, emr-6.6.0
	r6i.4xlarge	emr-5.35.0, emr-6.6.0
	r6i.8xlarge	emr-5.35.0, emr-6.6.0
	r6i.12xlarge	emr-5.35.0, emr-6.6.0
	r6i.16xlarge	emr-5.35.0, emr-6.6.0
	r6i.24xlarge	emr-5.35.0, emr-6.6.0
	r6i.32xlarge	emr-5.35.0, emr-6.6.0
	r6id.xlarge	emr-5.36.1, emr-6.8.0
	r6id.2xlarge	emr-5.36.1, emr-6.8.0

Kelas Instans	Tipe instans	Versi EMR Amazon yang didukung minimum (5.x, 6.x, 7.x)
	r6id.4xlarge	emr-5.36.1, emr-6.8.0
	r6id.8xlarge	emr-5.36.1, emr-6.8.0
	r6id.12xlarge	emr-5.36.1, emr-6.8.0
	r6id.16xlarge	emr-5.36.1, emr-6.8.0
	r6id.24xlarge	emr-5.36.1, emr-6.8.0
	r6id.32xlarge	emr-5.36.1, emr-6.8.0
	r6idn.xlarge	emr-5.36.1, emr-6.10.0
	r6idn.2xlarge	emr-5.36.1, emr-6.10.0
	r6idn.4xlarge	emr-5.36.1, emr-6.10.0
	r6idn.8xlarge	emr-5.36.1, emr-6.10.0
	r6idn.12xlarge	emr-5.36.1, emr-6.10.0
	r6idn.16xlarge	emr-5.36.1, emr-6.10.0
	r6idn.24xlarge	emr-5.36.1, emr-6.10.0
	r6idn.32xlarge	emr-5.36.1, emr-6.10.0
	r6in.xlarge	emr-5.36.1, emr-6.10.0
	r6in.2xlarge	emr-5.36.1, emr-6.10.0
	r6in.4xlarge	emr-5.36.1, emr-6.10.0
	r6in.8xlarge	emr-5.36.1, emr-6.10.0
	r6in.12xlarge	emr-5.36.1, emr-6.10.0

Kelas Instans	Tipe instans	Versi EMR Amazon yang didukung minimum (5.x, 6.x, 7.x)
	r6in.16xlarge	emr-5.36.1, emr-6.10.0
	r6in.24xlarge	emr-5.36.1, emr-6.10.0
	r6in.32xlarge	emr-5.36.1, emr-6.10.0
	r7g.xlarge	emr-5.36.1, emr-6.10.0, emr-7.0.0
	r7g.2xlarge	emr-5.36.1, emr-6.10.0, emr-7.0.0
	r7g.4xlarge	emr-5.36.1, emr-6.10.0, emr-7.0.0
	r7g.8xlarge	emr-5.36.1, emr-6.10.0, emr-7.0.0
	r7g.12xlarge	emr-5.36.1, emr-6.10.0, emr-7.0.0
	r7g.16xlarge	emr-5.36.1, emr-6.10.0, emr-7.0.0
	r7gd.xlarge	emr-5.36.1, emr-6.10.0, emr-7.0.0
	r7gd.2xlarge	emr-5.36.1, emr-6.10.0, emr-7.0.0
	r7gd.4xlarge	emr-5.36.1, emr-6.10.0, emr-7.0.0
	r7gd.8xlarge	emr-5.36.1, emr-6.10.0, emr-7.0.0

Kelas Instans	Tipe instans	Versi EMR Amazon yang didukung minimum (5.x, 6.x, 7.x)
	r7gd.12xlarge	emr-5.36.1, emr-6.10.0, emr-7.0.0
	r7gd.16xlarge	emr-5.36.1, emr-6.10.0, emr-7.0.0
	x1.16xlarge	emr-5.36.1, emr-6.9.0, emr-7.0.0
	x1.32xlarge	emr-5.36.1, emr-6.9.0, emr-7.0.0
	x1e.xlarge	emr-5.36.1, emr-6.10.0, emr-7.0.0
	x1e.2xlarge	emr-5.36.1, emr-6.10.0, emr-7.0.0
	x1e.4xlarge	emr-5.36.1, emr-6.10.0, emr-7.0.0
	x1e.8xlarge	emr-5.36.1, emr-6.10.0, emr-7.0.0
	x1e.16xlarge	emr-5.36.1, emr-6.10.0, emr-7.0.0
	x1e.32xlarge	emr-5.36.1, emr-6.10.0, emr-7.0.0
	x2idn.16xlarge	emr-5.36.1, emr-6.9.0
	x2idn.24xlarge	emr-5.36.1, emr-6.9.0
	x2idn.32xlarge	emr-5.36.1, emr-6.9.0

Kelas Instans	Tipe instans	Versi EMR Amazon yang didukung minimum (5.x, 6.x, 7.x)
	x2iedn.xlarge	emr-5.36.1, emr-6.9.0
	x2iedn.2xlarge	emr-5.36.1, emr-6.9.0
	x2iedn.4xlarge	emr-5.36.1, emr-6.9.0
	x2iedn.8xlarge	emr-5.36.1, emr-6.9.0
	x2iedn.16xlarge	emr-5.36.1, emr-6.9.0
	x2iedn.24xlarge	emr-5.36.1, emr-6.9.0
	x2iedn.32xlarge	emr-5.36.1, emr-6.9.0
	z1d.xlarge	emr-5.17.0, emr-6.0.0, emr-7.0.0
	z1d.2xlarge	emr-5.17.0, emr-6.0.0, emr-7.0.0
	z1d.3xlarge	emr-5.17.0, emr-6.0.0, emr-7.0.0
	z1d.6xlarge	emr-5.17.0, emr-6.0.0, emr-7.0.0
	z1d.12xlarge	emr-5.17.0, emr-6.0.0, emr-7.0.0
Penyimpanan Dioptimalkan	d3.xlarge	emr-5.33.0, emr-6.3.0, emr-7.0.0
	d3.2xlarge	emr-5.33.0, emr-6.3.0, emr-7.0.0

Kelas Instans	Tipe instans	Versi EMR Amazon yang didukung minimum (5.x, 6.x, 7.x)
	d3.4xlarge	emr-5.33.0, emr-6.3.0, emr-7.0.0
	d3.8xlarge	emr-5.33.0, emr-6.3.0, emr-7.0.0
	d3en.xlarge	emr-5.33.0, emr-6.3.0, emr-7.0.0
	d3en.2xlarge	emr-5.33.0, emr-6.3.0, emr-7.0.0
	d3en.4xlarge	emr-5.33.0, emr-6.3.0, emr-7.0.0
	d3en.6xlarge	emr-5.33.0, emr-6.3.0, emr-7.0.0
	d3en.8xlarge	emr-5.33.0, emr-6.3.0, emr-7.0.0
	d3en.12xlarge	emr-5.33.0, emr-6.3.0, emr-7.0.0
	i3.xlarge	emr-5.9.0, emr-6.0.0, emr-7.0.0
	i3.2xlarge	emr-5.9.0, emr-6.0.0, emr-7.0.0
	i3.4xlarge	emr-5.9.0, emr-6.0.0, emr-7.0.0
	i3.8xlarge	emr-5.9.0, emr-6.0.0, emr-7.0.0

Kelas Instans	Tipe instans	Versi EMR Amazon yang didukung minimum (5.x, 6.x, 7.x)
	i3.16xlarge	emr-5.9.0, emr-6.0.0, emr-7.0.0
	i3en.xlarge	emr-5.25.0, emr-6.0.0, emr-7.0.0
	i3en.2xlarge	emr-5.25.0, emr-6.0.0, emr-7.0.0
	i3en.3xlarge	emr-5.25.0, emr-6.0.0, emr-7.0.0
	i3en.6xlarge	emr-5.25.0, emr-6.0.0, emr-7.0.0
	i3en.12xlarge	emr-5.25.0, emr-6.0.0, emr-7.0.0
	i3en.24xlarge	emr-5.25.0, emr-6.0.0, emr-7.0.0
	i4i.xlarge	emr-5.36.1, emr-6.8.0
	i4i.2xlarge	emr-5.36.1, emr-6.8.0
	i4i.4xlarge	emr-5.36.1, emr-6.8.0
	i4i.8xlarge	emr-5.36.1, emr-6.8.0
	i4i.12xlarge	emr-5.36.1, emr-6.10.0
	i4i.16xlarge	emr-5.36.1, emr-6.8.0
	i4i.24xlarge	emr-5.36.1, emr-6.10.0
	i4i.32xlarge	emr-5.36.1, emr-6.8.0

Kelas Instans	Tipe instans	Versi EMR Amazon yang didukung minimum (5.x, 6.x, 7.x)
	im4gn.xlarge	emr-5.35.0, emr-6.6.0, emr-7.0.0
	im4gn.2xlarge	emr-5.35.0, emr-6.6.0, emr-7.0.0
	im4gn.4xlarge	emr-5.35.0, emr-6.6.0, emr-7.0.0
	im4gn.8xlarge	emr-5.35.0, emr-6.6.0, emr-7.0.0
	im4gn.16xlarge	emr-5.35.0, emr-6.6.0, emr-7.0.0
	is4gen.xlarge	emr-5.35.0, emr-6.6.0, emr-7.0.0
	is4gen.2xlarge	emr-5.35.0, emr-6.6.0, emr-7.0.0
	is4gen.4xlarge	emr-5.35.0, emr-6.6.0, emr-7.0.0
	is4gen.8xlarge	emr-5.35.0, emr-6.6.0, emr-7.0.0

Eropa (Zurich) - eu-central-2

Kelas Instans	Tipe instans	Versi EMR Amazon yang didukung minimum (5.x, 6.x, 7.x)
Tujuan Umum	m5.xlarge	emr-5.36.0, emr-6.7.0

Kelas Instans	Tipe instans	Versi EMR Amazon yang didukung minimum (5.x, 6.x, 7.x)
	m5.2xlarge	emr-5.36.0, emr-6.7.0
	m5.4xlarge	emr-5.36.0, emr-6.7.0
	m5.8xlarge	emr-5.36.0, emr-6.7.0
	m5.12xlarge	emr-5.36.0, emr-6.7.0
	m5.16xlarge	emr-5.36.0, emr-6.7.0
	m5.24xlarge	emr-5.36.0, emr-6.7.0
	m5d.xlarge	emr-5.36.0, emr-6.7.0
	m5d.2xlarge	emr-5.36.0, emr-6.7.0
	m5d.4xlarge	emr-5.36.0, emr-6.7.0
	m5d.8xlarge	emr-5.36.0, emr-6.7.0
	m5d.12xlarge	emr-5.36.0, emr-6.7.0
	m5d.16xlarge	emr-5.36.0, emr-6.7.0
	m5d.24xlarge	emr-5.36.0, emr-6.7.0
	m6g.xlarge	emr-5.36.0, emr-6.7.0
	m6g.2xlarge	emr-5.36.0, emr-6.7.0
	m6g.4xlarge	emr-5.36.0, emr-6.7.0
	m6g.8xlarge	emr-5.36.0, emr-6.7.0
	m6g.12xlarge	emr-5.36.0, emr-6.7.0
	m6g.16xlarge	emr-5.36.0, emr-6.7.0

Kelas Instans	Tipe instans	Versi EMR Amazon yang didukung minimum (5.x, 6.x, 7.x)
	m6gd.xlarge	emr-5.36.0, emr-6.7.0
	m6gd.2xlarge	emr-5.36.0, emr-6.7.0
	m6gd.4xlarge	emr-5.36.0, emr-6.7.0
	m6gd.8xlarge	emr-5.36.0, emr-6.7.0
	m6gd.12xlarge	emr-5.36.0, emr-6.7.0
	m6gd.16xlarge	emr-5.36.0, emr-6.7.0
	m6i.xlarge	emr-5.36.0, emr-6.7.0
	m6i.2xlarge	emr-5.36.0, emr-6.7.0
	m6i.4xlarge	emr-5.36.0, emr-6.7.0
	m6i.8xlarge	emr-5.36.0, emr-6.7.0
	m6i.12xlarge	emr-5.36.0, emr-6.7.0
	m6i.16xlarge	emr-5.36.0, emr-6.7.0
	m6i.24xlarge	emr-5.36.0, emr-6.7.0
	m6i.32xlarge	emr-5.36.0, emr-6.7.0
	m6id.xlarge	emr-5.36.1, emr-6.8.0
	m6id.2xlarge	emr-5.36.1, emr-6.8.0
	m6id.4xlarge	emr-5.36.1, emr-6.8.0
	m6id.8xlarge	emr-5.36.1, emr-6.8.0
	m6id.12xlarge	emr-5.36.1, emr-6.8.0

Kelas Instans	Tipe instans	Versi EMR Amazon yang didukung minimum (5.x, 6.x, 7.x)
	m6id.16xlarge	emr-5.36.1, emr-6.8.0
	m6id.24xlarge	emr-5.36.1, emr-6.8.0
	m6id.32xlarge	emr-5.36.1, emr-6.8.0
Komputasi Dioptimalkan	c5.xlarge	emr-5.36.0, emr-6.7.0
	c5.2xlarge	emr-5.36.0, emr-6.7.0
	c5.4xlarge	emr-5.36.0, emr-6.7.0
	c5.9xlarge	emr-5.36.0, emr-6.7.0
	c5.12xlarge	emr-5.36.0, emr-6.7.0
	c5.18xlarge	emr-5.36.0, emr-6.7.0
	c5.24xlarge	emr-5.36.0, emr-6.7.0
	c5d.xlarge	emr-5.36.0, emr-6.7.0
	c5d.2xlarge	emr-5.36.0, emr-6.7.0
	c5d.4xlarge	emr-5.36.0, emr-6.7.0
	c5d.9xlarge	emr-5.36.0, emr-6.7.0
	c5d.12xlarge	emr-5.36.0, emr-6.7.0
	c5d.18xlarge	emr-5.36.0, emr-6.7.0
	c5d.24xlarge	emr-5.36.0, emr-6.7.0
	c6g.xlarge	emr-5.36.0, emr-6.7.0
	c6g.2xlarge	emr-5.36.0, emr-6.7.0

Kelas Instans	Tipe instans	Versi EMR Amazon yang didukung minimum (5.x, 6.x, 7.x)
	c6g.4xlarge	emr-5.36.0, emr-6.7.0
	c6g.8xlarge	emr-5.36.0, emr-6.7.0
	c6g.12xlarge	emr-5.36.0, emr-6.7.0
	c6g.16xlarge	emr-5.36.0, emr-6.7.0
	c6gd.xlarge	emr-5.36.0, emr-6.7.0
	c6gd.2xlarge	emr-5.36.0, emr-6.7.0
	c6gd.4xlarge	emr-5.36.0, emr-6.7.0
	c6gd.8xlarge	emr-5.36.0, emr-6.7.0
	c6gd.12xlarge	emr-5.36.0, emr-6.7.0
	c6gd.16xlarge	emr-5.36.0, emr-6.7.0
	c6in.xlarge	emr-5.36.1, emr-6.10.0
	c6in.2xlarge	emr-5.36.1, emr-6.10.0
	c6in.4xlarge	emr-5.36.1, emr-6.10.0
	c6in.8xlarge	emr-5.36.1, emr-6.10.0
	c6in.12xlarge	emr-5.36.1, emr-6.10.0
	c6in.16xlarge	emr-5.36.1, emr-6.10.0
	c6in.24xlarge	emr-5.36.1, emr-6.10.0
	c6in.32xlarge	emr-5.36.1, emr-6.10.0
Memori Dioptimalkan	r5.xlarge	emr-5.36.0, emr-6.7.0

Kelas Instans	Tipe instans	Versi EMR Amazon yang didukung minimum (5.x, 6.x, 7.x)
	r5.2xlarge	emr-5.36.0, emr-6.7.0
	r5.4xlarge	emr-5.36.0, emr-6.7.0
	r5.8xlarge	emr-5.36.0, emr-6.7.0
	r5.12xlarge	emr-5.36.0, emr-6.7.0
	r5.16xlarge	emr-5.36.0, emr-6.7.0
	r5.24xlarge	emr-5.36.0, emr-6.7.0
	r5d.xlarge	emr-5.36.0, emr-6.7.0
	r5d.2xlarge	emr-5.36.0, emr-6.7.0
	r5d.4xlarge	emr-5.36.0, emr-6.7.0
	r5d.8xlarge	emr-5.36.0, emr-6.7.0
	r5d.12xlarge	emr-5.36.0, emr-6.7.0
	r5d.16xlarge	emr-5.36.0, emr-6.7.0
	r5d.24xlarge	emr-5.36.0, emr-6.7.0
	r6g.xlarge	emr-5.36.0, emr-6.7.0
	r6g.2xlarge	emr-5.36.0, emr-6.7.0
	r6g.4xlarge	emr-5.36.0, emr-6.7.0
	r6g.8xlarge	emr-5.36.0, emr-6.7.0
	r6g.12xlarge	emr-5.36.0, emr-6.7.0
	r6g.16xlarge	emr-5.36.0, emr-6.7.0

Kelas Instans	Tipe instans	Versi EMR Amazon yang didukung minimum (5.x, 6.x, 7.x)
	r6i.xlarge	emr-5.36.0, emr-6.7.0
	r6i.2xlarge	emr-5.36.0, emr-6.7.0
	r6i.4xlarge	emr-5.36.0, emr-6.7.0
	r6i.8xlarge	emr-5.36.0, emr-6.7.0
	r6i.12xlarge	emr-5.36.0, emr-6.7.0
	r6i.16xlarge	emr-5.36.0, emr-6.7.0
	r6i.24xlarge	emr-5.36.0, emr-6.7.0
	r6i.32xlarge	emr-5.36.0, emr-6.7.0
	x2idn.16xlarge	emr-5.36.1, emr-6.9.0
	x2idn.24xlarge	emr-5.36.1, emr-6.9.0
	x2idn.32xlarge	emr-5.36.1, emr-6.9.0
Penyimpanan Dioptimalkan	d3.xlarge	emr-5.36.0, emr-6.7.0
	d3.2xlarge	emr-5.36.0, emr-6.7.0
	d3.4xlarge	emr-5.36.0, emr-6.7.0
	d3.8xlarge	emr-5.36.0, emr-6.7.0
	i3.xlarge	emr-5.36.0, emr-6.7.0
	i3.2xlarge	emr-5.36.0, emr-6.7.0
	i3.4xlarge	emr-5.36.0, emr-6.7.0
	i3.8xlarge	emr-5.36.0, emr-6.7.0

Kelas Instans	Tipe instans	Versi EMR Amazon yang didukung minimum (5.x, 6.x, 7.x)
	i3.16xlarge	emr-5.36.0, emr-6.7.0
	i3en.xlarge	emr-5.36.0, emr-6.7.0
	i3en.2xlarge	emr-5.36.0, emr-6.7.0
	i3en.3xlarge	emr-5.36.0, emr-6.7.0
	i3en.6xlarge	emr-5.36.0, emr-6.7.0
	i3en.12xlarge	emr-5.36.0, emr-6.7.0
	i3en.24xlarge	emr-5.36.0, emr-6.7.0
	i4i.xlarge	emr-5.36.1, emr-6.8.0
	i4i.2xlarge	emr-5.36.1, emr-6.8.0
	i4i.4xlarge	emr-5.36.1, emr-6.8.0
	i4i.8xlarge	emr-5.36.1, emr-6.8.0
	i4i.12xlarge	emr-5.36.1, emr-6.10.0
	i4i.16xlarge	emr-5.36.1, emr-6.8.0
	i4i.24xlarge	emr-5.36.1, emr-6.10.0
	i4i.32xlarge	emr-5.36.1, emr-6.8.0

Europe (Ireland) - eu-west-1

Kelas Instans	Tipe instans	Versi EMR Amazon yang didukung minimum (5.x, 6.x, 7.x)
Tujuan Umum	m5.xlarge	emr-5.13.0, emr-6.0.0, emr-7.0.0
	m5.2xlarge	emr-5.13.0, emr-6.0.0, emr-7.0.0
	m5.4xlarge	emr-5.13.0, emr-6.0.0, emr-7.0.0
	m5.8xlarge	emr-5.13.0, emr-6.0.0, emr-7.0.0
	m5.12xlarge	emr-5.13.0, emr-6.0.0, emr-7.0.0
	m5.16xlarge	emr-5.13.0, emr-6.0.0, emr-7.0.0
	m5.24xlarge	emr-5.13.0, emr-6.0.0, emr-7.0.0
	m5a.xlarge	emr-5.20.0, emr-6.0.0, emr-7.0.0
	m5a.2xlarge	emr-5.20.0, emr-6.0.0, emr-7.0.0
	m5a.4xlarge	emr-5.20.0, emr-6.0.0, emr-7.0.0
	m5a.8xlarge	emr-5.20.0, emr-6.0.0, emr-7.0.0

Kelas Instans	Tipe instans	Versi EMR Amazon yang didukung minimum (5.x, 6.x, 7.x)
	m5a.12xlarge	emr-5.20.0, emr-6.0.0, emr-7.0.0
	m5a.16xlarge	emr-5.20.0, emr-6.0.0, emr-7.0.0
	m5a.24xlarge	emr-5.20.0, emr-6.0.0, emr-7.0.0
	m5ad.xlarge	emr-5.20.0, emr-6.0.0, emr-7.0.0
	m5ad.2xlarge	emr-5.20.0, emr-6.0.0, emr-7.0.0
	m5ad.4xlarge	emr-5.20.0, emr-6.0.0, emr-7.0.0
	m5ad.8xlarge	emr-5.33.0, emr-6.3.0, emr-7.0.0
	m5ad.12xlarge	emr-5.20.0, emr-6.0.0, emr-7.0.0
	m5ad.16xlarge	emr-5.33.0, emr-6.3.0, emr-7.0.0
	m5ad.24xlarge	emr-5.20.0, emr-6.0.0, emr-7.0.0
	m5d.xlarge	emr-5.13.0, emr-6.0.0, emr-7.0.0
	m5d.2xlarge	emr-5.13.0, emr-6.0.0, emr-7.0.0

Kelas Instans	Tipe instans	Versi EMR Amazon yang didukung minimum (5.x, 6.x, 7.x)
	m5d.4xlarge	emr-5.13.0, emr-6.0.0, emr-7.0.0
	m5d.8xlarge	emr-5.13.0, emr-6.0.0, emr-7.0.0
	m5d.12xlarge	emr-5.13.0, emr-6.0.0, emr-7.0.0
	m5d.16xlarge	emr-5.13.0, emr-6.0.0, emr-7.0.0
	m5d.24xlarge	emr-5.13.0, emr-6.0.0, emr-7.0.0
	m5dn.xlarge	emr-5.31.0, emr-6.1.0, emr-7.0.0
	m5dn.2xlarge	emr-5.31.0, emr-6.1.0, emr-7.0.0
	m5dn.4xlarge	emr-5.31.0, emr-6.1.0, emr-7.0.0
	m5dn.8xlarge	emr-5.31.0, emr-6.1.0, emr-7.0.0
	m5dn.12xlarge	emr-5.31.0, emr-6.1.0, emr-7.0.0
	m5dn.16xlarge	emr-5.31.0, emr-6.1.0, emr-7.0.0
	m5dn.24xlarge	emr-5.31.0, emr-6.1.0, emr-7.0.0

Kelas Instans	Tipe instans	Versi EMR Amazon yang didukung minimum (5.x, 6.x, 7.x)
	m5n.xlarge	emr-5.31.0, emr-6.1.0, emr-7.0.0
	m5n.2xlarge	emr-5.31.0, emr-6.1.0, emr-7.0.0
	m5n.4xlarge	emr-5.31.0, emr-6.1.0, emr-7.0.0
	m5n.8xlarge	emr-5.31.0, emr-6.1.0, emr-7.0.0
	m5n.12xlarge	emr-5.31.0, emr-6.1.0, emr-7.0.0
	m5n.16xlarge	emr-5.31.0, emr-6.1.0, emr-7.0.0
	m5n.24xlarge	emr-5.31.0, emr-6.1.0, emr-7.0.0
	m5zn.xlarge	emr-5.33.0, emr-6.3.0, emr-7.0.0
	m5zn.2xlarge	emr-5.33.0, emr-6.3.0, emr-7.0.0
	m5zn.3xlarge	emr-5.33.0, emr-6.3.0, emr-7.0.0
	m5zn.6xlarge	emr-5.33.0, emr-6.3.0, emr-7.0.0
	m5zn.12xlarge	emr-5.33.0, emr-6.3.0, emr-7.0.0

Kelas Instans	Tipe instans	Versi EMR Amazon yang didukung minimum (5.x, 6.x, 7.x)
	m6a.xlarge	emr-5.36.1, emr-6.8.0, emr-7.0.0
	m6a.2xlarge	emr-5.36.1, emr-6.8.0, emr-7.0.0
	m6a.4xlarge	emr-5.36.1, emr-6.8.0, emr-7.0.0
	m6a.8xlarge	emr-5.36.1, emr-6.8.0, emr-7.0.0
	m6a.12xlarge	emr-5.36.1, emr-6.8.0, emr-7.0.0
	m6a.16xlarge	emr-5.36.1, emr-6.8.0, emr-7.0.0
	m6a.24xlarge	emr-5.36.1, emr-6.8.0, emr-7.0.0
	m6a.32xlarge	emr-5.36.1, emr-6.8.0, emr-7.0.0
	m6a.48xlarge	emr-5.36.1, emr-6.8.0, emr-7.0.0
	m6g.xlarge	emr-5.30.0, emr-6.1.0, emr-7.0.0
	m6g.2xlarge	emr-5.30.0, emr-6.1.0, emr-7.0.0
	m6g.4xlarge	emr-5.30.0, emr-6.1.0, emr-7.0.0

Kelas Instans	Tipe instans	Versi EMR Amazon yang didukung minimum (5.x, 6.x, 7.x)
	m6g.8xlarge	emr-5.30.0, emr-6.1.0, emr-7.0.0
	m6g.12xlarge	emr-5.30.0, emr-6.1.0, emr-7.0.0
	m6g.16xlarge	emr-5.30.0, emr-6.1.0, emr-7.0.0
	m6gd.xlarge	emr-5.33.0, emr-6.3.0, emr-7.0.0
	m6gd.2xlarge	emr-5.33.0, emr-6.3.0, emr-7.0.0
	m6gd.4xlarge	emr-5.33.0, emr-6.3.0, emr-7.0.0
	m6gd.8xlarge	emr-5.33.0, emr-6.3.0, emr-7.0.0
	m6gd.12xlarge	emr-5.33.0, emr-6.3.0, emr-7.0.0
	m6gd.16xlarge	emr-5.33.0, emr-6.3.0, emr-7.0.0
	m6i.xlarge	emr-5.35.0, emr-6.6.0
	m6i.2xlarge	emr-5.35.0, emr-6.6.0
	m6i.4xlarge	emr-5.35.0, emr-6.6.0
	m6i.8xlarge	emr-5.35.0, emr-6.6.0
	m6i.12xlarge	emr-5.35.0, emr-6.6.0

Kelas Instans	Tipe instans	Versi EMR Amazon yang didukung minimum (5.x, 6.x, 7.x)
	m6i.16xlarge	emr-5.35.0, emr-6.6.0
	m6i.24xlarge	emr-5.35.0, emr-6.6.0
	m6i.32xlarge	emr-5.35.0, emr-6.6.0
	m6id.xlarge	emr-5.36.1, emr-6.8.0
	m6id.2xlarge	emr-5.36.1, emr-6.8.0
	m6id.4xlarge	emr-5.36.1, emr-6.8.0
	m6id.8xlarge	emr-5.36.1, emr-6.8.0
	m6id.12xlarge	emr-5.36.1, emr-6.8.0
	m6id.16xlarge	emr-5.36.1, emr-6.8.0
	m6id.24xlarge	emr-5.36.1, emr-6.8.0
	m6id.32xlarge	emr-5.36.1, emr-6.8.0
	m6idn.xlarge	emr-5.36.1, emr-6.10.0
	m6idn.2xlarge	emr-5.36.1, emr-6.10.0
	m6idn.4xlarge	emr-5.36.1, emr-6.10.0
	m6idn.8xlarge	emr-5.36.1, emr-6.10.0
	m6idn.12xlarge	emr-5.36.1, emr-6.10.0
	m6idn.16xlarge	emr-5.36.1, emr-6.10.0
	m6idn.24xlarge	emr-5.36.1, emr-6.10.0
	m6idn.32xlarge	emr-5.36.1, emr-6.10.0

Kelas Instans	Tipe instans	Versi EMR Amazon yang didukung minimum (5.x, 6.x, 7.x)
	m6in.xlarge	emr-5.36.1, emr-6.10.0
	m6in.2xlarge	emr-5.36.1, emr-6.10.0
	m6in.4xlarge	emr-5.36.1, emr-6.10.0
	m6in.8xlarge	emr-5.36.1, emr-6.10.0
	m6in.12xlarge	emr-5.36.1, emr-6.10.0
	m6in.16xlarge	emr-5.36.1, emr-6.10.0
	m6in.24xlarge	emr-5.36.1, emr-6.10.0
	m6in.32xlarge	emr-5.36.1, emr-6.10.0
	m7a.xlarge	emr-5.36.1, emr-6.10.0
	m7a.2xlarge	emr-5.36.1, emr-6.10.0
	m7a.4xlarge	emr-5.36.1, emr-6.10.0
	m7a.8xlarge	emr-5.36.1, emr-6.10.0
	m7a.12xlarge	emr-5.36.1, emr-6.10.0
	m7a.16xlarge	emr-5.36.1, emr-6.10.0
	m7a.24xlarge	emr-5.36.1, emr-6.10.0
	m7a.32xlarge	emr-5.36.1, emr-6.10.0
	m7a.48xlarge	emr-5.36.1, emr-6.10.0
	m7g.xlarge	emr-5.36.1, emr-6.10.0, emr-7.0.0

Kelas Instans	Tipe instans	Versi EMR Amazon yang didukung minimum (5.x, 6.x, 7.x)
	m7g.2xlarge	emr-5.36.1, emr-6.10.0, emr-7.0.0
	m7g.4xlarge	emr-5.36.1, emr-6.10.0, emr-7.0.0
	m7g.8xlarge	emr-5.36.1, emr-6.10.0, emr-7.0.0
	m7g.12xlarge	emr-5.36.1, emr-6.10.0, emr-7.0.0
	m7g.16xlarge	emr-5.36.1, emr-6.10.0, emr-7.0.0
	m7gd.xlarge	emr-5.36.1, emr-6.10.0, emr-7.0.0
	m7gd.2xlarge	emr-5.36.1, emr-6.10.0, emr-7.0.0
	m7gd.4xlarge	emr-5.36.1, emr-6.10.0, emr-7.0.0
	m7gd.8xlarge	emr-5.36.1, emr-6.10.0, emr-7.0.0
	m7gd.12xlarge	emr-5.36.1, emr-6.10.0, emr-7.0.0
	m7gd.16xlarge	emr-5.36.1, emr-6.10.0, emr-7.0.0
	m7i.xlarge	emr-5.36.1, emr-6.10.0
	m7i.2xlarge	emr-5.36.1, emr-6.10.0

Kelas Instans	Tipe instans	Versi EMR Amazon yang didukung minimum (5.x, 6.x, 7.x)
	m7i.4xlarge	emr-5.36.1, emr-6.10.0
	m7i.8xlarge	emr-5.36.1, emr-6.10.0
	m7i.12xlarge	emr-5.36.1, emr-6.10.0
	m7i.16xlarge	emr-5.36.1, emr-6.10.0
	m7i.24xlarge	emr-5.36.1, emr-6.10.0
	m7i.48xlarge	emr-5.36.1, emr-6.10.0
	m7i-flex.xlarge	emr-5.36.1, emr-6.10.0
	m7i-flex.2xlarge	emr-5.36.1, emr-6.10.0
	m7i-flex.4xlarge	emr-5.36.1, emr-6.10.0
	m7i-flex.8xlarge	emr-5.36.1, emr-6.10.0
Komputasi Dioptimalkan	c5.xlarge	emr-5.13.0, emr-6.0.0, emr-7.0.0
	c5.2xlarge	emr-5.13.0, emr-6.0.0, emr-7.0.0
	c5.4xlarge	emr-5.13.0, emr-6.0.0, emr-7.0.0
	c5.9xlarge	emr-5.13.0, emr-6.0.0, emr-7.0.0
	c5.12xlarge	emr-5.13.0, emr-6.0.0, emr-7.0.0
	c5.18xlarge	emr-5.13.0, emr-6.0.0, emr-7.0.0

Kelas Instans	Tipe instans	Versi EMR Amazon yang didukung minimum (5.x, 6.x, 7.x)
	c5.24xlarge	emr-5.13.0, emr-6.0.0, emr-7.0.0
	c5a.xlarge	emr-5.31.0, emr-6.1.0, emr-7.0.0
	c5a.2xlarge	emr-5.31.0, emr-6.1.0, emr-7.0.0
	c5a.4xlarge	emr-5.31.0, emr-6.1.0, emr-7.0.0
	c5a.8xlarge	emr-5.31.0, emr-6.1.0, emr-7.0.0
	c5a.12xlarge	emr-5.31.0, emr-6.1.0, emr-7.0.0
	c5a.16xlarge	emr-5.31.0, emr-6.1.0, emr-7.0.0
	c5a.24xlarge	emr-5.31.0, emr-6.1.0, emr-7.0.0
	c5ad.xlarge	emr-5.33.0, emr-6.3.0, emr-7.0.0
	c5ad.2xlarge	emr-5.33.0, emr-6.3.0, emr-7.0.0
	c5ad.4xlarge	emr-5.33.0, emr-6.3.0, emr-7.0.0
	c5ad.8xlarge	emr-5.33.0, emr-6.3.0, emr-7.0.0

Kelas Instans	Tipe instans	Versi EMR Amazon yang didukung minimum (5.x, 6.x, 7.x)
	c5ad.12xlarge	emr-5.33.0, emr-6.3.0, emr-7.0.0
	c5ad.16xlarge	emr-5.33.0, emr-6.3.0, emr-7.0.0
	c5ad.24xlarge	emr-5.33.0, emr-6.3.0, emr-7.0.0
	c5d.xlarge	emr-5.13.0, emr-6.0.0, emr-7.0.0
	c5d.2xlarge	emr-5.13.0, emr-6.0.0, emr-7.0.0
	c5d.4xlarge	emr-5.13.0, emr-6.0.0, emr-7.0.0
	c5d.9xlarge	emr-5.13.0, emr-6.0.0, emr-7.0.0
	c5d.12xlarge	emr-5.13.0, emr-6.0.0, emr-7.0.0
	c5d.18xlarge	emr-5.13.0, emr-6.0.0, emr-7.0.0
	c5d.24xlarge	emr-5.13.0, emr-6.0.0, emr-7.0.0
	c5n.xlarge	emr-5.20.0, emr-6.0.0, emr-7.0.0
	c5n.2xlarge	emr-5.20.0, emr-6.0.0, emr-7.0.0

Kelas Instans	Tipe instans	Versi EMR Amazon yang didukung minimum (5.x, 6.x, 7.x)
	c5n.4xlarge	emr-5.20.0, emr-6.0.0, emr-7.0.0
	c5n.9xlarge	emr-5.20.0, emr-6.0.0, emr-7.0.0
	c5n.18xlarge	emr-5.20.0, emr-6.0.0, emr-7.0.0
	c6a.xlarge	emr-5.36.1, emr-6.8.0, emr-7.0.0
	c6a.2xlarge	emr-5.36.1, emr-6.8.0, emr-7.0.0
	c6a.4xlarge	emr-5.36.1, emr-6.8.0, emr-7.0.0
	c6a.8xlarge	emr-5.36.1, emr-6.8.0, emr-7.0.0
	c6a.12xlarge	emr-5.36.1, emr-6.8.0, emr-7.0.0
	c6a.16xlarge	emr-5.36.1, emr-6.8.0, emr-7.0.0
	c6a.24xlarge	emr-5.36.1, emr-6.8.0, emr-7.0.0
	c6a.32xlarge	emr-5.36.1, emr-6.8.0, emr-7.0.0
	c6a.48xlarge	emr-5.36.1, emr-6.8.0, emr-7.0.0

Kelas Instans	Tipe instans	Versi EMR Amazon yang didukung minimum (5.x, 6.x, 7.x)
	c6g.xlarge	emr-5.31.0, emr-6.1.0, emr-7.0.0
	c6g.2xlarge	emr-5.31.0, emr-6.1.0, emr-7.0.0
	c6g.4xlarge	emr-5.31.0, emr-6.1.0, emr-7.0.0
	c6g.8xlarge	emr-5.31.0, emr-6.1.0, emr-7.0.0
	c6g.12xlarge	emr-5.31.0, emr-6.1.0, emr-7.0.0
	c6g.16xlarge	emr-5.31.0, emr-6.1.0, emr-7.0.0
	c6gd.xlarge	emr-5.33.0, emr-6.3.0, emr-7.0.0
	c6gd.2xlarge	emr-5.33.0, emr-6.3.0, emr-7.0.0
	c6gd.4xlarge	emr-5.33.0, emr-6.3.0, emr-7.0.0
	c6gd.8xlarge	emr-5.33.0, emr-6.3.0, emr-7.0.0
	c6gd.12xlarge	emr-5.33.0, emr-6.3.0, emr-7.0.0
	c6gd.16xlarge	emr-5.33.0, emr-6.3.0, emr-7.0.0

Kelas Instans	Tipe instans	Versi EMR Amazon yang didukung minimum (5.x, 6.x, 7.x)
	c6gn.xlarge	emr-5.33.0, emr-6.3.0, emr-7.0.0
	c6gn.2xlarge	emr-5.33.0, emr-6.3.0, emr-7.0.0
	c6gn.4xlarge	emr-5.33.0, emr-6.3.0, emr-7.0.0
	c6gn.8xlarge	emr-5.33.0, emr-6.3.0, emr-7.0.0
	c6gn.12xlarge	emr-5.33.0, emr-6.3.0, emr-7.0.0
	c6gn.16xlarge	emr-5.33.0, emr-6.3.0, emr-7.0.0
	c6i.xlarge	emr-5.35.0, emr-6.6.0
	c6i.2xlarge	emr-5.35.0, emr-6.6.0
	c6i.4xlarge	emr-5.35.0, emr-6.6.0
	c6i.8xlarge	emr-5.35.0, emr-6.6.0
	c6i.12xlarge	emr-5.35.0, emr-6.6.0
	c6i.16xlarge	emr-5.35.0, emr-6.6.0
	c6i.24xlarge	emr-5.35.0, emr-6.6.0
	c6i.32xlarge	emr-5.35.0, emr-6.6.0
	c6id.xlarge	emr-5.36.1, emr-6.8.0
	c6id.2xlarge	emr-5.36.1, emr-6.8.0

Kelas Instans	Tipe instans	Versi EMR Amazon yang didukung minimum (5.x, 6.x, 7.x)
	c6id.4xlarge	emr-5.36.1, emr-6.8.0
	c6id.8xlarge	emr-5.36.1, emr-6.8.0
	c6id.12xlarge	emr-5.36.1, emr-6.8.0
	c6id.16xlarge	emr-5.36.1, emr-6.8.0
	c6id.24xlarge	emr-5.36.1, emr-6.8.0
	c6id.32xlarge	emr-5.36.1, emr-6.8.0
	c6in.xlarge	emr-5.36.1, emr-6.10.0
	c6in.2xlarge	emr-5.36.1, emr-6.10.0
	c6in.4xlarge	emr-5.36.1, emr-6.10.0
	c6in.8xlarge	emr-5.36.1, emr-6.10.0
	c6in.12xlarge	emr-5.36.1, emr-6.10.0
	c6in.16xlarge	emr-5.36.1, emr-6.10.0
	c6in.24xlarge	emr-5.36.1, emr-6.10.0
	c6in.32xlarge	emr-5.36.1, emr-6.10.0
	c7a.xlarge	emr-5.36.1, emr-6.10.0
	c7a.2xlarge	emr-5.36.1, emr-6.10.0
	c7a.4xlarge	emr-5.36.1, emr-6.10.0
	c7a.8xlarge	emr-5.36.1, emr-6.10.0
	c7a.12xlarge	emr-5.36.1, emr-6.10.0

Kelas Instans	Tipe instans	Versi EMR Amazon yang didukung minimum (5.x, 6.x, 7.x)
	c7a.16xlarge	emr-5.36.1, emr-6.10.0
	c7a.24xlarge	emr-5.36.1, emr-6.10.0
	c7a.32xlarge	emr-5.36.1, emr-6.10.0
	c7a.48xlarge	emr-5.36.1, emr-6.10.0
	c7g.xlarge	emr-5.36.1, emr-6.7.0, emr-7.0.0
	c7g.2xlarge	emr-5.36.1, emr-6.7.0, emr-7.0.0
	c7g.4xlarge	emr-5.36.1, emr-6.7.0, emr-7.0.0
	c7g.8xlarge	emr-5.36.1, emr-6.7.0, emr-7.0.0
	c7g.12xlarge	emr-5.36.1, emr-6.7.0, emr-7.0.0
	c7g.16xlarge	emr-5.36.1, emr-6.7.0, emr-7.0.0
	c7gd.xlarge	emr-5.36.1, emr-6.10.0, emr-7.0.0
	c7gd.2xlarge	emr-5.36.1, emr-6.10.0, emr-7.0.0
	c7gd.4xlarge	emr-5.36.1, emr-6.10.0, emr-7.0.0

Kelas Instans	Tipe instans	Versi EMR Amazon yang didukung minimum (5.x, 6.x, 7.x)
	c7gd.8xlarge	emr-5.36.1, emr-6.10.0, emr-7.0.0
	c7gd.12xlarge	emr-5.36.1, emr-6.10.0, emr-7.0.0
	c7gd.16xlarge	emr-5.36.1, emr-6.10.0, emr-7.0.0
	c7gn.xlarge	emr-5.36.1, emr-6.10.0, emr-7.0.0
	c7gn.2xlarge	emr-5.36.1, emr-6.10.0, emr-7.0.0
	c7gn.4xlarge	emr-5.36.1, emr-6.10.0, emr-7.0.0
	c7gn.8xlarge	emr-5.36.1, emr-6.10.0, emr-7.0.0
	c7gn.12xlarge	emr-5.36.1, emr-6.10.0, emr-7.0.0
	c7gn.16xlarge	emr-5.36.1, emr-6.10.0, emr-7.0.0
	c7i.xlarge	emr-5.36.1, emr-6.10.0
	c7i.2xlarge	emr-5.36.1, emr-6.10.0
	c7i.4xlarge	emr-5.36.1, emr-6.10.0
	c7i.8xlarge	emr-5.36.1, emr-6.10.0
	c7i.12xlarge	emr-5.36.1, emr-6.10.0

Kelas Instans	Tipe instans	Versi EMR Amazon yang didukung minimum (5.x, 6.x, 7.x)
	c7i.16xlarge	emr-5.36.1, emr-6.10.0
	c7i.24xlarge	emr-5.36.1, emr-6.10.0
	c7i.48xlarge	emr-5.36.1, emr-6.10.0
Komputasi yang Dipercepat	g3.4xlarge	emr-5.18.0, emr-6.0.0, emr-7.0.0
	g3.8xlarge	emr-5.18.0, emr-6.0.0, emr-7.0.0
	g3.16xlarge	emr-5.18.0, emr-6.0.0, emr-7.0.0
	g3s.xlarge	emr-5.19.0, emr-6.0.0, emr-7.0.0
	g4dn.xlarge	emr-5.30.0, emr-6.0.0, emr-7.0.0
	g4dn.2xlarge	emr-5.30.0, emr-6.0.0, emr-7.0.0
	g4dn.4xlarge	emr-5.30.0, emr-6.0.0, emr-7.0.0
	g4dn.8xlarge	emr-5.30.0, emr-6.0.0, emr-7.0.0
	g4dn.12xlarge	emr-5.30.0, emr-6.0.0, emr-7.0.0
	g4dn.16xlarge	emr-5.30.0, emr-6.0.0, emr-7.0.0

Kelas Instans	Tipe instans	Versi EMR Amazon yang didukung minimum (5.x, 6.x, 7.x)
	g5.xlarge	emr-5.36.1, emr-6.9.0, emr-7.0.0
	g5.2xlarge	emr-5.36.1, emr-6.9.0, emr-7.0.0
	g5.4xlarge	emr-5.36.1, emr-6.9.0, emr-7.0.0
	g5.8xlarge	emr-5.36.1, emr-6.9.0, emr-7.0.0
	g5.12xlarge	emr-5.36.1, emr-6.9.0, emr-7.0.0
	g5.16xlarge	emr-5.36.1, emr-6.9.0, emr-7.0.0
	g5.24xlarge	emr-5.36.1, emr-6.9.0, emr-7.0.0
	g5.48xlarge	emr-5.36.1, emr-6.9.0, emr-7.0.0
	p2.xlarge	emr-5.10.0, emr-6.0.0, emr-7.0.0
	p2.8xlarge	emr-5.10.0, emr-6.0.0, emr-7.0.0
	p2.16xlarge	emr-5.10.0, emr-6.0.0, emr-7.0.0
	p3.2xlarge	emr-5.10.0, emr-6.0.0, emr-7.0.0

Kelas Instans	Tipe instans	Versi EMR Amazon yang didukung minimum (5.x, 6.x, 7.x)
	p3.8xlarge	emr-5.10.0, emr-6.0.0, emr-7.0.0
	p3.16xlarge	emr-5.10.0, emr-6.0.0, emr-7.0.0
Memori Dioptimalkan	r5.xlarge	emr-5.13.0, emr-6.0.0, emr-7.0.0
	r5.2xlarge	emr-5.13.0, emr-6.0.0, emr-7.0.0
	r5.4xlarge	emr-5.13.0, emr-6.0.0, emr-7.0.0
	r5.8xlarge	emr-5.13.0, emr-6.0.0, emr-7.0.0
	r5.12xlarge	emr-5.13.0, emr-6.0.0, emr-7.0.0
	r5.16xlarge	emr-5.13.0, emr-6.0.0, emr-7.0.0
	r5.24xlarge	emr-5.13.0, emr-6.0.0, emr-7.0.0
	r5a.xlarge	emr-5.20.0, emr-6.0.0, emr-7.0.0
	r5a.2xlarge	emr-5.20.0, emr-6.0.0, emr-7.0.0
	r5a.4xlarge	emr-5.20.0, emr-6.0.0, emr-7.0.0

Kelas Instans	Tipe instans	Versi EMR Amazon yang didukung minimum (5.x, 6.x, 7.x)
	r5a.8xlarge	emr-5.20.0, emr-6.0.0, emr-7.0.0
	r5a.12xlarge	emr-5.20.0, emr-6.0.0, emr-7.0.0
	r5a.16xlarge	emr-5.20.0, emr-6.0.0, emr-7.0.0
	r5a.24xlarge	emr-5.20.0, emr-6.0.0, emr-7.0.0
	r5ad.xlarge	emr-5.20.0, emr-6.0.0, emr-7.0.0
	r5ad.2xlarge	emr-5.20.0, emr-6.0.0, emr-7.0.0
	r5ad.4xlarge	emr-5.20.0, emr-6.0.0, emr-7.0.0
	r5ad.8xlarge	emr-5.33.0, emr-6.3.0, emr-7.0.0
	r5ad.12xlarge	emr-5.20.0, emr-6.0.0, emr-7.0.0
	r5ad.16xlarge	emr-5.33.0, emr-6.3.0, emr-7.0.0
	r5ad.24xlarge	emr-5.20.0, emr-6.0.0, emr-7.0.0
	r5b.xlarge	emr-5.33.0, emr-6.3.0, emr-7.0.0

Kelas Instans	Tipe instans	Versi EMR Amazon yang didukung minimum (5.x, 6.x, 7.x)
	r5b.2xlarge	emr-5.33.0, emr-6.3.0, emr-7.0.0
	r5b.4xlarge	emr-5.33.0, emr-6.3.0, emr-7.0.0
	r5b.8xlarge	emr-5.33.0, emr-6.3.0, emr-7.0.0
	r5b.12xlarge	emr-5.33.0, emr-6.3.0, emr-7.0.0
	r5b.16xlarge	emr-5.33.0, emr-6.3.0, emr-7.0.0
	r5b.24xlarge	emr-5.33.0, emr-6.3.0, emr-7.0.0
	r5d.xlarge	emr-5.13.0, emr-6.0.0, emr-7.0.0
	r5d.2xlarge	emr-5.13.0, emr-6.0.0, emr-7.0.0
	r5d.4xlarge	emr-5.13.0, emr-6.0.0, emr-7.0.0
	r5d.8xlarge	emr-5.13.0, emr-6.0.0, emr-7.0.0
	r5d.12xlarge	emr-5.13.0, emr-6.0.0, emr-7.0.0
	r5d.16xlarge	emr-5.13.0, emr-6.0.0, emr-7.0.0

Kelas Instans	Tipe instans	Versi EMR Amazon yang didukung minimum (5.x, 6.x, 7.x)
	r5d.24xlarge	emr-5.13.0, emr-6.0.0, emr-7.0.0
	r5dn.xlarge	emr-5.31.0, emr-6.1.0, emr-7.0.0
	r5dn.2xlarge	emr-5.31.0, emr-6.1.0, emr-7.0.0
	r5dn.4xlarge	emr-5.31.0, emr-6.1.0, emr-7.0.0
	r5dn.8xlarge	emr-5.31.0, emr-6.1.0, emr-7.0.0
	r5dn.12xlarge	emr-5.31.0, emr-6.1.0, emr-7.0.0
	r5dn.16xlarge	emr-5.31.0, emr-6.1.0, emr-7.0.0
	r5dn.24xlarge	emr-5.31.0, emr-6.1.0, emr-7.0.0
	r5n.xlarge	emr-5.31.0, emr-6.1.0, emr-7.0.0
	r5n.2xlarge	emr-5.31.0, emr-6.1.0, emr-7.0.0
	r5n.4xlarge	emr-5.31.0, emr-6.1.0, emr-7.0.0
	r5n.8xlarge	emr-5.31.0, emr-6.1.0, emr-7.0.0

Kelas Instans	Tipe instans	Versi EMR Amazon yang didukung minimum (5.x, 6.x, 7.x)
	r5n.12xlarge	emr-5.31.0, emr-6.1.0, emr-7.0.0
	r5n.16xlarge	emr-5.31.0, emr-6.1.0, emr-7.0.0
	r5n.24xlarge	emr-5.31.0, emr-6.1.0, emr-7.0.0
	r6a.xlarge	emr-5.36.1, emr-6.8.0, emr-7.0.0
	r6a.2xlarge	emr-5.36.1, emr-6.8.0, emr-7.0.0
	r6a.4xlarge	emr-5.36.1, emr-6.8.0, emr-7.0.0
	r6a.8xlarge	emr-5.36.1, emr-6.8.0, emr-7.0.0
	r6a.12xlarge	emr-5.36.1, emr-6.8.0, emr-7.0.0
	r6a.16xlarge	emr-5.36.1, emr-6.8.0, emr-7.0.0
	r6a.24xlarge	emr-5.36.1, emr-6.8.0, emr-7.0.0
	r6a.32xlarge	emr-5.36.1, emr-6.8.0, emr-7.0.0
	r6a.48xlarge	emr-5.36.1, emr-6.8.0, emr-7.0.0

Kelas Instans	Tipe instans	Versi EMR Amazon yang didukung minimum (5.x, 6.x, 7.x)
	r6g.xlarge	emr-5.31.0, emr-6.1.0, emr-7.0.0
	r6g.2xlarge	emr-5.31.0, emr-6.1.0, emr-7.0.0
	r6g.4xlarge	emr-5.31.0, emr-6.1.0, emr-7.0.0
	r6g.8xlarge	emr-5.31.0, emr-6.1.0, emr-7.0.0
	r6g.12xlarge	emr-5.31.0, emr-6.1.0, emr-7.0.0
	r6g.16xlarge	emr-5.31.0, emr-6.1.0, emr-7.0.0
	r6gd.xlarge	emr-5.33.0, emr-6.3.0, emr-7.0.0
	r6gd.2xlarge	emr-5.33.0, emr-6.3.0, emr-7.0.0
	r6gd.4xlarge	emr-5.33.0, emr-6.3.0, emr-7.0.0
	r6gd.8xlarge	emr-5.33.0, emr-6.3.0, emr-7.0.0
	r6gd.12xlarge	emr-5.33.0, emr-6.3.0, emr-7.0.0
	r6gd.16xlarge	emr-5.33.0, emr-6.3.0, emr-7.0.0

Kelas Instans	Tipe instans	Versi EMR Amazon yang didukung minimum (5.x, 6.x, 7.x)
	r6i.xlarge	emr-5.35.0, emr-6.6.0
	r6i.2xlarge	emr-5.35.0, emr-6.6.0
	r6i.4xlarge	emr-5.35.0, emr-6.6.0
	r6i.8xlarge	emr-5.35.0, emr-6.6.0
	r6i.12xlarge	emr-5.35.0, emr-6.6.0
	r6i.16xlarge	emr-5.35.0, emr-6.6.0
	r6i.24xlarge	emr-5.35.0, emr-6.6.0
	r6i.32xlarge	emr-5.35.0, emr-6.6.0
	r6id.xlarge	emr-5.36.1, emr-6.8.0
	r6id.2xlarge	emr-5.36.1, emr-6.8.0
	r6id.4xlarge	emr-5.36.1, emr-6.8.0
	r6id.8xlarge	emr-5.36.1, emr-6.8.0
	r6id.12xlarge	emr-5.36.1, emr-6.8.0
	r6id.16xlarge	emr-5.36.1, emr-6.8.0
	r6id.24xlarge	emr-5.36.1, emr-6.8.0
	r6id.32xlarge	emr-5.36.1, emr-6.8.0
	r6idn.xlarge	emr-5.36.1, emr-6.10.0
	r6idn.2xlarge	emr-5.36.1, emr-6.10.0
	r6idn.4xlarge	emr-5.36.1, emr-6.10.0

Kelas Instans	Tipe instans	Versi EMR Amazon yang didukung minimum (5.x, 6.x, 7.x)
	r6idn.8xlarge	emr-5.36.1, emr-6.10.0
	r6idn.12xlarge	emr-5.36.1, emr-6.10.0
	r6idn.16xlarge	emr-5.36.1, emr-6.10.0
	r6idn.24xlarge	emr-5.36.1, emr-6.10.0
	r6idn.32xlarge	emr-5.36.1, emr-6.10.0
	r6in.xlarge	emr-5.36.1, emr-6.10.0
	r6in.2xlarge	emr-5.36.1, emr-6.10.0
	r6in.4xlarge	emr-5.36.1, emr-6.10.0
	r6in.8xlarge	emr-5.36.1, emr-6.10.0
	r6in.12xlarge	emr-5.36.1, emr-6.10.0
	r6in.16xlarge	emr-5.36.1, emr-6.10.0
	r6in.24xlarge	emr-5.36.1, emr-6.10.0
	r6in.32xlarge	emr-5.36.1, emr-6.10.0
	r7a.xlarge	emr-5.36.1, emr-6.10.0
	r7a.2xlarge	emr-5.36.1, emr-6.10.0
	r7a.4xlarge	emr-5.36.1, emr-6.10.0
	r7a.8xlarge	emr-5.36.1, emr-6.10.0
	r7a.12xlarge	emr-5.36.1, emr-6.10.0
	r7a.16xlarge	emr-5.36.1, emr-6.10.0

Kelas Instans	Tipe instans	Versi EMR Amazon yang didukung minimum (5.x, 6.x, 7.x)
	r7a.24xlarge	emr-5.36.1, emr-6.10.0
	r7a.32xlarge	emr-5.36.1, emr-6.10.0
	r7a.48xlarge	emr-5.36.1, emr-6.10.0
	r7g.xlarge	emr-5.36.1, emr-6.10.0, emr-7.0.0
	r7g.2xlarge	emr-5.36.1, emr-6.10.0, emr-7.0.0
	r7g.4xlarge	emr-5.36.1, emr-6.10.0, emr-7.0.0
	r7g.8xlarge	emr-5.36.1, emr-6.10.0, emr-7.0.0
	r7g.12xlarge	emr-5.36.1, emr-6.10.0, emr-7.0.0
	r7g.16xlarge	emr-5.36.1, emr-6.10.0, emr-7.0.0
	r7gd.xlarge	emr-5.36.1, emr-6.10.0, emr-7.0.0
	r7gd.2xlarge	emr-5.36.1, emr-6.10.0, emr-7.0.0
	r7gd.4xlarge	emr-5.36.1, emr-6.10.0, emr-7.0.0
	r7gd.8xlarge	emr-5.36.1, emr-6.10.0, emr-7.0.0

Kelas Instans	Tipe instans	Versi EMR Amazon yang didukung minimum (5.x, 6.x, 7.x)
	r7gd.12xlarge	emr-5.36.1, emr-6.10.0, emr-7.0.0
	r7gd.16xlarge	emr-5.36.1, emr-6.10.0, emr-7.0.0
	r7i.xlarge	emr-5.36.1, emr-6.10.0
	r7i.2xlarge	emr-5.36.1, emr-6.10.0
	r7i.4xlarge	emr-5.36.1, emr-6.10.0
	r7i.8xlarge	emr-5.36.1, emr-6.10.0
	r7i.12xlarge	emr-5.36.1, emr-6.10.0
	r7i.16xlarge	emr-5.36.1, emr-6.10.0
	r7i.24xlarge	emr-5.36.1, emr-6.10.0
	r7i.48xlarge	emr-5.36.1, emr-6.10.0
	x1.16xlarge	emr-5.36.1, emr-6.9.0, emr-7.0.0
	x1.32xlarge	emr-5.36.1, emr-6.9.0, emr-7.0.0
	x1e.xlarge	emr-5.36.1, emr-6.10.0, emr-7.0.0
	x1e.2xlarge	emr-5.36.1, emr-6.10.0, emr-7.0.0
	x1e.4xlarge	emr-5.36.1, emr-6.10.0, emr-7.0.0

Kelas Instans	Tipe instans	Versi EMR Amazon yang didukung minimum (5.x, 6.x, 7.x)
	x1e.8xlarge	emr-5.36.1, emr-6.10.0, emr-7.0.0
	x1e.16xlarge	emr-5.36.1, emr-6.10.0, emr-7.0.0
	x1e.32xlarge	emr-5.36.1, emr-6.10.0, emr-7.0.0
	x2gd.xlarge	emr-5.36.1, emr-6.10.0, emr-7.0.0
	x2gd.2xlarge	emr-5.36.1, emr-6.10.0, emr-7.0.0
	x2gd.4xlarge	emr-5.36.1, emr-6.10.0, emr-7.0.0
	x2gd.8xlarge	emr-5.36.1, emr-6.10.0, emr-7.0.0
	x2gd.12xlarge	emr-5.36.1, emr-6.10.0, emr-7.0.0
	x2gd.16xlarge	emr-5.36.1, emr-6.10.0, emr-7.0.0
	x2idn.16xlarge	emr-5.36.1, emr-6.9.0
	x2idn.24xlarge	emr-5.36.1, emr-6.9.0
	x2idn.32xlarge	emr-5.36.1, emr-6.9.0
	x2iedn.xlarge	emr-5.36.1, emr-6.9.0
	x2iedn.2xlarge	emr-5.36.1, emr-6.9.0

Kelas Instans	Tipe instans	Versi EMR Amazon yang didukung minimum (5.x, 6.x, 7.x)
	x2iedn.4xlarge	emr-5.36.1, emr-6.9.0
	x2iedn.8xlarge	emr-5.36.1, emr-6.9.0
	x2iedn.16xlarge	emr-5.36.1, emr-6.9.0
	x2iedn.24xlarge	emr-5.36.1, emr-6.9.0
	x2iedn.32xlarge	emr-5.36.1, emr-6.9.0
	z1d.xlarge	emr-5.17.0, emr-6.0.0, emr-7.0.0
	z1d.2xlarge	emr-5.17.0, emr-6.0.0, emr-7.0.0
	z1d.3xlarge	emr-5.17.0, emr-6.0.0, emr-7.0.0
	z1d.6xlarge	emr-5.17.0, emr-6.0.0, emr-7.0.0
	z1d.12xlarge	emr-5.17.0, emr-6.0.0, emr-7.0.0
Penyimpanan Dioptimalkan	d3.xlarge	emr-5.33.0, emr-6.3.0, emr-7.0.0
	d3.2xlarge	emr-5.33.0, emr-6.3.0, emr-7.0.0
	d3.4xlarge	emr-5.33.0, emr-6.3.0, emr-7.0.0
	d3.8xlarge	emr-5.33.0, emr-6.3.0, emr-7.0.0

Kelas Instans	Tipe instans	Versi EMR Amazon yang didukung minimum (5.x, 6.x, 7.x)
	d3en.xlarge	emr-5.33.0, emr-6.3.0, emr-7.0.0
	d3en.2xlarge	emr-5.33.0, emr-6.3.0, emr-7.0.0
	d3en.4xlarge	emr-5.33.0, emr-6.3.0, emr-7.0.0
	d3en.6xlarge	emr-5.33.0, emr-6.3.0, emr-7.0.0
	d3en.8xlarge	emr-5.33.0, emr-6.3.0, emr-7.0.0
	d3en.12xlarge	emr-5.33.0, emr-6.3.0, emr-7.0.0
	h1.2xlarge	emr-5.17.0, emr-6.0.0, emr-7.0.0
	h1.4xlarge	emr-5.17.0, emr-6.0.0, emr-7.0.0
	h1.8xlarge	emr-5.17.0, emr-6.0.0, emr-7.0.0
	h1.16xlarge	emr-5.17.0, emr-6.0.0, emr-7.0.0
	i3.xlarge	emr-5.9.0, emr-6.0.0, emr-7.0.0
	i3.2xlarge	emr-5.9.0, emr-6.0.0, emr-7.0.0

Kelas Instans	Tipe instans	Versi EMR Amazon yang didukung minimum (5.x, 6.x, 7.x)
	i3.4xlarge	emr-5.9.0, emr-6.0.0, emr-7.0.0
	i3.8xlarge	emr-5.9.0, emr-6.0.0, emr-7.0.0
	i3.16xlarge	emr-5.9.0, emr-6.0.0, emr-7.0.0
	i3en.xlarge	emr-5.25.0, emr-6.0.0, emr-7.0.0
	i3en.2xlarge	emr-5.25.0, emr-6.0.0, emr-7.0.0
	i3en.3xlarge	emr-5.25.0, emr-6.0.0, emr-7.0.0
	i3en.6xlarge	emr-5.25.0, emr-6.0.0, emr-7.0.0
	i3en.12xlarge	emr-5.25.0, emr-6.0.0, emr-7.0.0
	i3en.24xlarge	emr-5.25.0, emr-6.0.0, emr-7.0.0
	i4g.xlarge	emr-5.36.1, emr-6.10.0, emr-7.0.0
	i4g.2xlarge	emr-5.36.1, emr-6.10.0, emr-7.0.0
	i4g.4xlarge	emr-5.36.1, emr-6.10.0, emr-7.0.0

Kelas Instans	Tipe instans	Versi EMR Amazon yang didukung minimum (5.x, 6.x, 7.x)
	i4g.8xlarge	emr-5.36.1, emr-6.10.0, emr-7.0.0
	i4g.16xlarge	emr-5.36.1, emr-6.10.0, emr-7.0.0
	i4i.xlarge	emr-5.36.1, emr-6.8.0
	i4i.2xlarge	emr-5.36.1, emr-6.8.0
	i4i.4xlarge	emr-5.36.1, emr-6.8.0
	i4i.8xlarge	emr-5.36.1, emr-6.8.0
	i4i.12xlarge	emr-5.36.1, emr-6.10.0
	i4i.16xlarge	emr-5.36.1, emr-6.8.0
	i4i.24xlarge	emr-5.36.1, emr-6.10.0
	i4i.32xlarge	emr-5.36.1, emr-6.8.0
	im4gn.xlarge	emr-5.35.0, emr-6.6.0, emr-7.0.0
	im4gn.2xlarge	emr-5.35.0, emr-6.6.0, emr-7.0.0
	im4gn.4xlarge	emr-5.35.0, emr-6.6.0, emr-7.0.0
	im4gn.8xlarge	emr-5.35.0, emr-6.6.0, emr-7.0.0
	im4gn.16xlarge	emr-5.35.0, emr-6.6.0, emr-7.0.0

Kelas Instans	Tipe instans	Versi EMR Amazon yang didukung minimum (5.x, 6.x, 7.x)
	is4gen.xlarge	emr-5.35.0, emr-6.6.0, emr-7.0.0
	is4gen.2xlarge	emr-5.35.0, emr-6.6.0, emr-7.0.0
	is4gen.4xlarge	emr-5.35.0, emr-6.6.0, emr-7.0.0
	is4gen.8xlarge	emr-5.35.0, emr-6.6.0, emr-7.0.0

Europe (London) - eu-west-2

Kelas Instans	Tipe instans	Versi EMR Amazon yang didukung minimum (5.x, 6.x, 7.x)
Tujuan Umum	m5.xlarge	emr-5.13.0, emr-6.0.0
	m5.2xlarge	emr-5.13.0, emr-6.0.0
	m5.4xlarge	emr-5.13.0, emr-6.0.0
	m5.8xlarge	emr-5.13.0, emr-6.0.0
	m5.12xlarge	emr-5.13.0, emr-6.0.0
	m5.16xlarge	emr-5.13.0, emr-6.0.0
	m5.24xlarge	emr-5.13.0, emr-6.0.0
	m5a.xlarge	emr-5.20.0, emr-6.0.0
	m5a.2xlarge	emr-5.20.0, emr-6.0.0

Kelas Instans	Tipe instans	Versi EMR Amazon yang didukung minimum (5.x, 6.x, 7.x)
	m5a.4xlarge	emr-5.20.0, emr-6.0.0
	m5a.8xlarge	emr-5.20.0, emr-6.0.0
	m5a.12xlarge	emr-5.20.0, emr-6.0.0
	m5a.16xlarge	emr-5.20.0, emr-6.0.0
	m5a.24xlarge	emr-5.20.0, emr-6.0.0
	m5ad.xlarge	emr-5.20.0, emr-6.0.0
	m5ad.2xlarge	emr-5.20.0, emr-6.0.0
	m5ad.4xlarge	emr-5.20.0, emr-6.0.0
	m5ad.8xlarge	emr-5.33.0, emr-6.3.0
	m5ad.12xlarge	emr-5.20.0, emr-6.0.0
	m5ad.16xlarge	emr-5.33.0, emr-6.3.0
	m5ad.24xlarge	emr-5.20.0, emr-6.0.0
	m5d.xlarge	emr-5.13.0, emr-6.0.0
	m5d.2xlarge	emr-5.13.0, emr-6.0.0
	m5d.4xlarge	emr-5.13.0, emr-6.0.0
	m5d.8xlarge	emr-5.13.0, emr-6.0.0
	m5d.12xlarge	emr-5.13.0, emr-6.0.0
	m5d.16xlarge	emr-5.13.0, emr-6.0.0
	m5d.24xlarge	emr-5.13.0, emr-6.0.0

Kelas Instans	Tipe instans	Versi EMR Amazon yang didukung minimum (5.x, 6.x, 7.x)
	m6a.xlarge	emr-5.36.1, emr-6.8.0
	m6a.2xlarge	emr-5.36.1, emr-6.8.0
	m6a.4xlarge	emr-5.36.1, emr-6.8.0
	m6a.8xlarge	emr-5.36.1, emr-6.8.0
	m6a.12xlarge	emr-5.36.1, emr-6.8.0
	m6a.16xlarge	emr-5.36.1, emr-6.8.0
	m6a.24xlarge	emr-5.36.1, emr-6.8.0
	m6a.32xlarge	emr-5.36.1, emr-6.8.0
	m6a.48xlarge	emr-5.36.1, emr-6.8.0
	m6g.xlarge	emr-5.30.0, emr-6.1.0
	m6g.2xlarge	emr-5.30.0, emr-6.1.0
	m6g.4xlarge	emr-5.30.0, emr-6.1.0
	m6g.8xlarge	emr-5.30.0, emr-6.1.0
	m6g.12xlarge	emr-5.30.0, emr-6.1.0
	m6g.16xlarge	emr-5.30.0, emr-6.1.0
	m6gd.xlarge	emr-5.33.0, emr-6.3.0
	m6gd.2xlarge	emr-5.33.0, emr-6.3.0
	m6gd.4xlarge	emr-5.33.0, emr-6.3.0
	m6gd.8xlarge	emr-5.33.0, emr-6.3.0

Kelas Instans	Tipe instans	Versi EMR Amazon yang didukung minimum (5.x, 6.x, 7.x)
	m6gd.12xlarge	emr-5.33.0, emr-6.3.0
	m6gd.16xlarge	emr-5.33.0, emr-6.3.0
	m6i.xlarge	emr-5.35.0, emr-6.6.0
	m6i.2xlarge	emr-5.35.0, emr-6.6.0
	m6i.4xlarge	emr-5.35.0, emr-6.6.0
	m6i.8xlarge	emr-5.35.0, emr-6.6.0
	m6i.12xlarge	emr-5.35.0, emr-6.6.0
	m6i.16xlarge	emr-5.35.0, emr-6.6.0
	m6i.24xlarge	emr-5.35.0, emr-6.6.0
	m6i.32xlarge	emr-5.35.0, emr-6.6.0
Komputasi Dioptimalkan	c5.xlarge	emr-5.13.0, emr-6.0.0
	c5.2xlarge	emr-5.13.0, emr-6.0.0
	c5.4xlarge	emr-5.13.0, emr-6.0.0
	c5.9xlarge	emr-5.13.0, emr-6.0.0
	c5.12xlarge	emr-5.13.0, emr-6.0.0
	c5.18xlarge	emr-5.13.0, emr-6.0.0
	c5.24xlarge	emr-5.13.0, emr-6.0.0
	c5a.xlarge	emr-5.31.0, emr-6.1.0
	c5a.2xlarge	emr-5.31.0, emr-6.1.0

Kelas Instans	Tipe instans	Versi EMR Amazon yang didukung minimum (5.x, 6.x, 7.x)
	c5a.4xlarge	emr-5.31.0, emr-6.1.0
	c5a.8xlarge	emr-5.31.0, emr-6.1.0
	c5a.12xlarge	emr-5.31.0, emr-6.1.0
	c5a.16xlarge	emr-5.31.0, emr-6.1.0
	c5a.24xlarge	emr-5.31.0, emr-6.1.0
	c5d.xlarge	emr-5.13.0, emr-6.0.0
	c5d.2xlarge	emr-5.13.0, emr-6.0.0
	c5d.4xlarge	emr-5.13.0, emr-6.0.0
	c5d.9xlarge	emr-5.13.0, emr-6.0.0
	c5d.12xlarge	emr-5.13.0, emr-6.0.0
	c5d.18xlarge	emr-5.13.0, emr-6.0.0
	c5d.24xlarge	emr-5.13.0, emr-6.0.0
	c5n.xlarge	emr-5.20.0, emr-6.0.0
	c5n.2xlarge	emr-5.20.0, emr-6.0.0
	c5n.4xlarge	emr-5.20.0, emr-6.0.0
	c5n.9xlarge	emr-5.20.0, emr-6.0.0
	c5n.18xlarge	emr-5.20.0, emr-6.0.0
	c6a.xlarge	emr-5.36.1, emr-6.8.0
	c6a.2xlarge	emr-5.36.1, emr-6.8.0

Kelas Instans	Tipe instans	Versi EMR Amazon yang didukung minimum (5.x, 6.x, 7.x)
	c6a.4xlarge	emr-5.36.1, emr-6.8.0
	c6a.8xlarge	emr-5.36.1, emr-6.8.0
	c6a.12xlarge	emr-5.36.1, emr-6.8.0
	c6a.16xlarge	emr-5.36.1, emr-6.8.0
	c6a.24xlarge	emr-5.36.1, emr-6.8.0
	c6a.32xlarge	emr-5.36.1, emr-6.8.0
	c6a.48xlarge	emr-5.36.1, emr-6.8.0
	c6g.xlarge	emr-5.31.0, emr-6.1.0
	c6g.2xlarge	emr-5.31.0, emr-6.1.0
	c6g.4xlarge	emr-5.31.0, emr-6.1.0
	c6g.8xlarge	emr-5.31.0, emr-6.1.0
	c6g.12xlarge	emr-5.31.0, emr-6.1.0
	c6g.16xlarge	emr-5.31.0, emr-6.1.0
	c6gd.xlarge	emr-5.33.0, emr-6.3.0
	c6gd.2xlarge	emr-5.33.0, emr-6.3.0
	c6gd.4xlarge	emr-5.33.0, emr-6.3.0
	c6gd.8xlarge	emr-5.33.0, emr-6.3.0
	c6gd.12xlarge	emr-5.33.0, emr-6.3.0
	c6gd.16xlarge	emr-5.33.0, emr-6.3.0

Kelas Instans	Tipe instans	Versi EMR Amazon yang didukung minimum (5.x, 6.x, 7.x)
	c6gn.xlarge	emr-5.33.0, emr-6.3.0
	c6gn.2xlarge	emr-5.33.0, emr-6.3.0
	c6gn.4xlarge	emr-5.33.0, emr-6.3.0
	c6gn.8xlarge	emr-5.33.0, emr-6.3.0
	c6gn.12xlarge	emr-5.33.0, emr-6.3.0
	c6gn.16xlarge	emr-5.33.0, emr-6.3.0
	c6i.xlarge	emr-5.35.0, emr-6.6.0
	c6i.2xlarge	emr-5.35.0, emr-6.6.0
	c6i.4xlarge	emr-5.35.0, emr-6.6.0
	c6i.8xlarge	emr-5.35.0, emr-6.6.0
	c6i.12xlarge	emr-5.35.0, emr-6.6.0
	c6i.16xlarge	emr-5.35.0, emr-6.6.0
	c6i.24xlarge	emr-5.35.0, emr-6.6.0
	c6i.32xlarge	emr-5.35.0, emr-6.6.0
	c6id.xlarge	emr-5.36.1, emr-6.8.0
	c6id.2xlarge	emr-5.36.1, emr-6.8.0
	c6id.4xlarge	emr-5.36.1, emr-6.8.0
	c6id.8xlarge	emr-5.36.1, emr-6.8.0
	c6id.12xlarge	emr-5.36.1, emr-6.8.0

Kelas Instans	Tipe instans	Versi EMR Amazon yang didukung minimum (5.x, 6.x, 7.x)
	c6id.16xlarge	emr-5.36.1, emr-6.8.0
	c6id.24xlarge	emr-5.36.1, emr-6.8.0
	c6id.32xlarge	emr-5.36.1, emr-6.8.0
	c6in.xlarge	emr-5.36.1, emr-6.10.0
	c6in.2xlarge	emr-5.36.1, emr-6.10.0
	c6in.4xlarge	emr-5.36.1, emr-6.10.0
	c6in.8xlarge	emr-5.36.1, emr-6.10.0
	c6in.12xlarge	emr-5.36.1, emr-6.10.0
	c6in.16xlarge	emr-5.36.1, emr-6.10.0
	c6in.24xlarge	emr-5.36.1, emr-6.10.0
	c6in.32xlarge	emr-5.36.1, emr-6.10.0
Komputasi yang Dipercepat	g3.4xlarge	emr-5.18.0, emr-6.0.0
	g3.8xlarge	emr-5.18.0, emr-6.0.0
	g3.16xlarge	emr-5.18.0, emr-6.0.0
	g3s.xlarge	emr-5.19.0, emr-6.0.0
	g4dn.xlarge	emr-5.30.0, emr-6.0.0
	g4dn.2xlarge	emr-5.30.0, emr-6.0.0
	g4dn.4xlarge	emr-5.30.0, emr-6.0.0
	g4dn.8xlarge	emr-5.30.0, emr-6.0.0

Kelas Instans	Tipe instans	Versi EMR Amazon yang didukung minimum (5.x, 6.x, 7.x)
	g4dn.12xlarge	emr-5.30.0, emr-6.0.0
	g4dn.16xlarge	emr-5.30.0, emr-6.0.0
	g5.xlarge	emr-5.36.1, emr-6.9.0
	g5.2xlarge	emr-5.36.1, emr-6.9.0
	g5.4xlarge	emr-5.36.1, emr-6.9.0
	g5.8xlarge	emr-5.36.1, emr-6.9.0
	g5.12xlarge	emr-5.36.1, emr-6.9.0
	g5.16xlarge	emr-5.36.1, emr-6.9.0
	g5.24xlarge	emr-5.36.1, emr-6.9.0
	g5.48xlarge	emr-5.36.1, emr-6.9.0
	p3.2xlarge	emr-5.10.0, emr-6.0.0
	p3.8xlarge	emr-5.10.0, emr-6.0.0
	p3.16xlarge	emr-5.10.0, emr-6.0.0
Memori Dioptimalkan	r5.xlarge	emr-5.13.0, emr-6.0.0
	r5.2xlarge	emr-5.13.0, emr-6.0.0
	r5.4xlarge	emr-5.13.0, emr-6.0.0
	r5.8xlarge	emr-5.13.0, emr-6.0.0
	r5.12xlarge	emr-5.13.0, emr-6.0.0
	r5.16xlarge	emr-5.13.0, emr-6.0.0

Kelas Instans	Tipe instans	Versi EMR Amazon yang didukung minimum (5.x, 6.x, 7.x)
	r5.24xlarge	emr-5.13.0, emr-6.0.0
	r5a.xlarge	emr-5.20.0, emr-6.0.0
	r5a.2xlarge	emr-5.20.0, emr-6.0.0
	r5a.4xlarge	emr-5.20.0, emr-6.0.0
	r5a.8xlarge	emr-5.20.0, emr-6.0.0
	r5a.12xlarge	emr-5.20.0, emr-6.0.0
	r5a.16xlarge	emr-5.20.0, emr-6.0.0
	r5a.24xlarge	emr-5.20.0, emr-6.0.0
	r5ad.xlarge	emr-5.20.0, emr-6.0.0
	r5ad.2xlarge	emr-5.20.0, emr-6.0.0
	r5ad.4xlarge	emr-5.20.0, emr-6.0.0
	r5ad.8xlarge	emr-5.33.0, emr-6.3.0
	r5ad.12xlarge	emr-5.20.0, emr-6.0.0
	r5ad.16xlarge	emr-5.33.0, emr-6.3.0
	r5ad.24xlarge	emr-5.20.0, emr-6.0.0
	r5b.xlarge	emr-5.33.0, emr-6.3.0
	r5b.2xlarge	emr-5.33.0, emr-6.3.0
	r5b.4xlarge	emr-5.33.0, emr-6.3.0
	r5b.8xlarge	emr-5.33.0, emr-6.3.0

Kelas Instans	Tipe instans	Versi EMR Amazon yang didukung minimum (5.x, 6.x, 7.x)
	r5b.12xlarge	emr-5.33.0, emr-6.3.0
	r5b.16xlarge	emr-5.33.0, emr-6.3.0
	r5b.24xlarge	emr-5.33.0, emr-6.3.0
	r5d.xlarge	emr-5.13.0, emr-6.0.0
	r5d.2xlarge	emr-5.13.0, emr-6.0.0
	r5d.4xlarge	emr-5.13.0, emr-6.0.0
	r5d.8xlarge	emr-5.13.0, emr-6.0.0
	r5d.12xlarge	emr-5.13.0, emr-6.0.0
	r5d.16xlarge	emr-5.13.0, emr-6.0.0
	r5d.24xlarge	emr-5.13.0, emr-6.0.0
	r5n.xlarge	emr-5.31.0, emr-6.1.0
	r5n.2xlarge	emr-5.31.0, emr-6.1.0
	r5n.4xlarge	emr-5.31.0, emr-6.1.0
	r5n.8xlarge	emr-5.31.0, emr-6.1.0
	r5n.12xlarge	emr-5.31.0, emr-6.1.0
	r5n.16xlarge	emr-5.31.0, emr-6.1.0
	r5n.24xlarge	emr-5.31.0, emr-6.1.0
	r6g.xlarge	emr-5.31.0, emr-6.1.0
	r6g.2xlarge	emr-5.31.0, emr-6.1.0

Kelas Instans	Tipe instans	Versi EMR Amazon yang didukung minimum (5.x, 6.x, 7.x)
	r6g.4xlarge	emr-5.31.0, emr-6.1.0
	r6g.8xlarge	emr-5.31.0, emr-6.1.0
	r6g.12xlarge	emr-5.31.0, emr-6.1.0
	r6g.16xlarge	emr-5.31.0, emr-6.1.0
	r6gd.xlarge	emr-5.33.0, emr-6.3.0
	r6gd.2xlarge	emr-5.33.0, emr-6.3.0
	r6gd.4xlarge	emr-5.33.0, emr-6.3.0
	r6gd.8xlarge	emr-5.33.0, emr-6.3.0
	r6gd.12xlarge	emr-5.33.0, emr-6.3.0
	r6gd.16xlarge	emr-5.33.0, emr-6.3.0
	r6i.xlarge	emr-5.35.0, emr-6.6.0
	r6i.2xlarge	emr-5.35.0, emr-6.6.0
	r6i.4xlarge	emr-5.35.0, emr-6.6.0
	r6i.8xlarge	emr-5.35.0, emr-6.6.0
	r6i.12xlarge	emr-5.35.0, emr-6.6.0
	r6i.16xlarge	emr-5.35.0, emr-6.6.0
	r6i.24xlarge	emr-5.35.0, emr-6.6.0
	r6i.32xlarge	emr-5.35.0, emr-6.6.0
	r6id.xlarge	emr-5.36.1, emr-6.8.0

Kelas Instans	Tipe instans	Versi EMR Amazon yang didukung minimum (5.x, 6.x, 7.x)
	r6id.2xlarge	emr-5.36.1, emr-6.8.0
	r6id.4xlarge	emr-5.36.1, emr-6.8.0
	r6id.8xlarge	emr-5.36.1, emr-6.8.0
	r6id.12xlarge	emr-5.36.1, emr-6.8.0
	r6id.16xlarge	emr-5.36.1, emr-6.8.0
	r6id.24xlarge	emr-5.36.1, emr-6.8.0
	r6id.32xlarge	emr-5.36.1, emr-6.8.0
	x1.16xlarge	emr-5.36.1, emr-6.9.0
	x1.32xlarge	emr-5.36.1, emr-6.9.0
	x2idn.16xlarge	emr-5.36.1, emr-6.9.0
	x2idn.24xlarge	emr-5.36.1, emr-6.9.0
	x2idn.32xlarge	emr-5.36.1, emr-6.9.0
	x2iedn.xlarge	emr-5.36.1, emr-6.9.0
	x2iedn.2xlarge	emr-5.36.1, emr-6.9.0
	x2iedn.4xlarge	emr-5.36.1, emr-6.9.0
	x2iedn.8xlarge	emr-5.36.1, emr-6.9.0
	x2iedn.16xlarge	emr-5.36.1, emr-6.9.0
	x2iedn.24xlarge	emr-5.36.1, emr-6.9.0
	x2iedn.32xlarge	emr-5.36.1, emr-6.9.0

Kelas Instans	Tipe instans	Versi EMR Amazon yang didukung minimum (5.x, 6.x, 7.x)
	z1d.xlarge	emr-5.17.0, emr-6.0.0
	z1d.2xlarge	emr-5.17.0, emr-6.0.0
	z1d.3xlarge	emr-5.17.0, emr-6.0.0
	z1d.6xlarge	emr-5.17.0, emr-6.0.0
	z1d.12xlarge	emr-5.17.0, emr-6.0.0
Penyimpanan Dioptimalkan	d3.xlarge	emr-5.33.0, emr-6.3.0
	d3.2xlarge	emr-5.33.0, emr-6.3.0
	d3.4xlarge	emr-5.33.0, emr-6.3.0
	d3.8xlarge	emr-5.33.0, emr-6.3.0
	i3.xlarge	emr-5.9.0, emr-6.0.0
	i3.2xlarge	emr-5.9.0, emr-6.0.0
	i3.4xlarge	emr-5.9.0, emr-6.0.0
	i3.8xlarge	emr-5.9.0, emr-6.0.0
	i3.16xlarge	emr-5.9.0, emr-6.0.0
	i3en.xlarge	emr-5.25.0, emr-6.0.0
	i3en.2xlarge	emr-5.25.0, emr-6.0.0
	i3en.3xlarge	emr-5.25.0, emr-6.0.0
	i3en.6xlarge	emr-5.25.0, emr-6.0.0
	i3en.12xlarge	emr-5.25.0, emr-6.0.0

Kelas Instans	Tipe instans	Versi EMR Amazon yang didukung minimum (5.x, 6.x, 7.x)
	i3en.24xlarge	emr-5.25.0, emr-6.0.0
	i4i.xlarge	emr-5.36.1, emr-6.8.0
	i4i.2xlarge	emr-5.36.1, emr-6.8.0
	i4i.4xlarge	emr-5.36.1, emr-6.8.0
	i4i.8xlarge	emr-5.36.1, emr-6.8.0
	i4i.12xlarge	emr-5.36.1, emr-6.10.0
	i4i.16xlarge	emr-5.36.1, emr-6.8.0
	i4i.24xlarge	emr-5.36.1, emr-6.10.0
	i4i.32xlarge	emr-5.36.1, emr-6.8.0
	im4gn.xlarge	emr-5.35.0, emr-6.6.0
	im4gn.2xlarge	emr-5.35.0, emr-6.6.0
	im4gn.4xlarge	emr-5.35.0, emr-6.6.0
	im4gn.8xlarge	emr-5.35.0, emr-6.6.0
	im4gn.16xlarge	emr-5.35.0, emr-6.6.0
	is4gen.xlarge	emr-5.35.0, emr-6.6.0
	is4gen.2xlarge	emr-5.35.0, emr-6.6.0
	is4gen.4xlarge	emr-5.35.0, emr-6.6.0
	is4gen.8xlarge	emr-5.35.0, emr-6.6.0

Eropa (Milan) - eu-selatan-1

Kelas Instans	Tipe instans	Versi EMR Amazon yang didukung minimum (5.x, 6.x, 7.x)
Tujuan Umum	m5.xlarge	emr-5.29.0, emr-6.0.0, emr-7.0.0
	m5.2xlarge	emr-5.29.0, emr-6.0.0, emr-7.0.0
	m5.4xlarge	emr-5.29.0, emr-6.0.0, emr-7.0.0
	m5.8xlarge	emr-5.29.0, emr-6.0.0, emr-7.0.0
	m5.12xlarge	emr-5.29.0, emr-6.0.0, emr-7.0.0
	m5.16xlarge	emr-5.29.0, emr-6.0.0, emr-7.0.0
	m5.24xlarge	emr-5.29.0, emr-6.0.0, emr-7.0.0
	m5a.xlarge	emr-5.29.0, emr-6.0.0, emr-7.0.0
	m5a.2xlarge	emr-5.29.0, emr-6.0.0, emr-7.0.0
	m5a.4xlarge	emr-5.29.0, emr-6.0.0, emr-7.0.0
	m5a.8xlarge	emr-5.29.0, emr-6.0.0, emr-7.0.0

Kelas Instans	Tipe instans	Versi EMR Amazon yang didukung minimum (5.x, 6.x, 7.x)
	m5a.12xlarge	emr-5.29.0, emr-6.0.0, emr-7.0.0
	m5a.16xlarge	emr-5.29.0, emr-6.0.0, emr-7.0.0
	m5a.24xlarge	emr-5.29.0, emr-6.0.0, emr-7.0.0
	m5d.xlarge	emr-5.29.0, emr-6.0.0, emr-7.0.0
	m5d.2xlarge	emr-5.29.0, emr-6.0.0, emr-7.0.0
	m5d.4xlarge	emr-5.29.0, emr-6.0.0, emr-7.0.0
	m5d.8xlarge	emr-5.29.0, emr-6.0.0, emr-7.0.0
	m5d.12xlarge	emr-5.29.0, emr-6.0.0, emr-7.0.0
	m5d.16xlarge	emr-5.29.0, emr-6.0.0, emr-7.0.0
	m5d.24xlarge	emr-5.29.0, emr-6.0.0, emr-7.0.0
	m6a.xlarge	emr-5.36.1, emr-6.8.0, emr-7.0.0
	m6a.2xlarge	emr-5.36.1, emr-6.8.0, emr-7.0.0

Kelas Instans	Tipe instans	Versi EMR Amazon yang didukung minimum (5.x, 6.x, 7.x)
	m6a.4xlarge	emr-5.36.1, emr-6.8.0, emr-7.0.0
	m6a.8xlarge	emr-5.36.1, emr-6.8.0, emr-7.0.0
	m6a.12xlarge	emr-5.36.1, emr-6.8.0, emr-7.0.0
	m6a.16xlarge	emr-5.36.1, emr-6.8.0, emr-7.0.0
	m6a.24xlarge	emr-5.36.1, emr-6.8.0, emr-7.0.0
	m6a.32xlarge	emr-5.36.1, emr-6.8.0, emr-7.0.0
	m6a.48xlarge	emr-5.36.1, emr-6.8.0, emr-7.0.0
	m6g.xlarge	emr-5.30.0, emr-6.1.0, emr-7.0.0
	m6g.2xlarge	emr-5.30.0, emr-6.1.0, emr-7.0.0
	m6g.4xlarge	emr-5.30.0, emr-6.1.0, emr-7.0.0
	m6g.8xlarge	emr-5.30.0, emr-6.1.0, emr-7.0.0
	m6g.12xlarge	emr-5.30.0, emr-6.1.0, emr-7.0.0

Kelas Instans	Tipe instans	Versi EMR Amazon yang didukung minimum (5.x, 6.x, 7.x)
	m6g.16xlarge	emr-5.30.0, emr-6.1.0, emr-7.0.0
	m6i.xlarge	emr-5.35.0, emr-6.6.0
	m6i.2xlarge	emr-5.35.0, emr-6.6.0
	m6i.4xlarge	emr-5.35.0, emr-6.6.0
	m6i.8xlarge	emr-5.35.0, emr-6.6.0
	m6i.12xlarge	emr-5.35.0, emr-6.6.0
	m6i.16xlarge	emr-5.35.0, emr-6.6.0
	m6i.24xlarge	emr-5.35.0, emr-6.6.0
	m6i.32xlarge	emr-5.35.0, emr-6.6.0
Komputasi Dioptimalkan	c5.xlarge	emr-5.29.0, emr-6.0.0, emr-7.0.0
	c5.2xlarge	emr-5.29.0, emr-6.0.0, emr-7.0.0
	c5.4xlarge	emr-5.29.0, emr-6.0.0, emr-7.0.0
	c5.9xlarge	emr-5.29.0, emr-6.0.0, emr-7.0.0
	c5.12xlarge	emr-5.29.0, emr-6.0.0, emr-7.0.0
	c5.18xlarge	emr-5.29.0, emr-6.0.0, emr-7.0.0

Kelas Instans	Tipe instans	Versi EMR Amazon yang didukung minimum (5.x, 6.x, 7.x)
	c5.24xlarge	emr-5.29.0, emr-6.0.0, emr-7.0.0
	c5a.xlarge	emr-5.31.0, emr-6.1.0, emr-7.0.0
	c5a.2xlarge	emr-5.31.0, emr-6.1.0, emr-7.0.0
	c5a.4xlarge	emr-5.31.0, emr-6.1.0, emr-7.0.0
	c5a.8xlarge	emr-5.31.0, emr-6.1.0, emr-7.0.0
	c5a.12xlarge	emr-5.31.0, emr-6.1.0, emr-7.0.0
	c5a.16xlarge	emr-5.31.0, emr-6.1.0, emr-7.0.0
	c5a.24xlarge	emr-5.31.0, emr-6.1.0, emr-7.0.0
	c5ad.xlarge	emr-5.33.0, emr-6.3.0, emr-7.0.0
	c5ad.2xlarge	emr-5.33.0, emr-6.3.0, emr-7.0.0
	c5ad.4xlarge	emr-5.33.0, emr-6.3.0, emr-7.0.0
	c5ad.8xlarge	emr-5.33.0, emr-6.3.0, emr-7.0.0

Kelas Instans	Tipe instans	Versi EMR Amazon yang didukung minimum (5.x, 6.x, 7.x)
	c5ad.12xlarge	emr-5.33.0, emr-6.3.0, emr-7.0.0
	c5ad.16xlarge	emr-5.33.0, emr-6.3.0, emr-7.0.0
	c5ad.24xlarge	emr-5.33.0, emr-6.3.0, emr-7.0.0
	c5d.xlarge	emr-5.29.0, emr-6.0.0, emr-7.0.0
	c5d.2xlarge	emr-5.29.0, emr-6.0.0, emr-7.0.0
	c5d.4xlarge	emr-5.29.0, emr-6.0.0, emr-7.0.0
	c5d.9xlarge	emr-5.29.0, emr-6.0.0, emr-7.0.0
	c5d.12xlarge	emr-5.29.0, emr-6.0.0, emr-7.0.0
	c5d.18xlarge	emr-5.29.0, emr-6.0.0, emr-7.0.0
	c5d.24xlarge	emr-5.29.0, emr-6.0.0, emr-7.0.0
	c5n.xlarge	emr-5.29.0, emr-6.0.0, emr-7.0.0
	c5n.2xlarge	emr-5.29.0, emr-6.0.0, emr-7.0.0

Kelas Instans	Tipe instans	Versi EMR Amazon yang didukung minimum (5.x, 6.x, 7.x)
	c5n.4xlarge	emr-5.29.0, emr-6.0.0, emr-7.0.0
	c5n.9xlarge	emr-5.29.0, emr-6.0.0, emr-7.0.0
	c5n.18xlarge	emr-5.29.0, emr-6.0.0, emr-7.0.0
	c6g.xlarge	emr-5.31.0, emr-6.1.0, emr-7.0.0
	c6g.2xlarge	emr-5.31.0, emr-6.1.0, emr-7.0.0
	c6g.4xlarge	emr-5.31.0, emr-6.1.0, emr-7.0.0
	c6g.8xlarge	emr-5.31.0, emr-6.1.0, emr-7.0.0
	c6g.12xlarge	emr-5.31.0, emr-6.1.0, emr-7.0.0
	c6g.16xlarge	emr-5.31.0, emr-6.1.0, emr-7.0.0
	c6gn.xlarge	emr-5.33.0, emr-6.3.0, emr-7.0.0
	c6gn.2xlarge	emr-5.33.0, emr-6.3.0, emr-7.0.0
	c6gn.4xlarge	emr-5.33.0, emr-6.3.0, emr-7.0.0

Kelas Instans	Tipe instans	Versi EMR Amazon yang didukung minimum (5.x, 6.x, 7.x)
	c6gn.8xlarge	emr-5.33.0, emr-6.3.0, emr-7.0.0
	c6gn.12xlarge	emr-5.33.0, emr-6.3.0, emr-7.0.0
	c6gn.16xlarge	emr-5.33.0, emr-6.3.0, emr-7.0.0
	c6i.xlarge	emr-5.35.0, emr-6.6.0
	c6i.2xlarge	emr-5.35.0, emr-6.6.0
	c6i.4xlarge	emr-5.35.0, emr-6.6.0
	c6i.8xlarge	emr-5.35.0, emr-6.6.0
	c6i.12xlarge	emr-5.35.0, emr-6.6.0
	c6i.16xlarge	emr-5.35.0, emr-6.6.0
	c6i.24xlarge	emr-5.35.0, emr-6.6.0
	c6i.32xlarge	emr-5.35.0, emr-6.6.0
	c6in.xlarge	emr-5.36.1, emr-6.10.0
	c6in.2xlarge	emr-5.36.1, emr-6.10.0
	c6in.4xlarge	emr-5.36.1, emr-6.10.0
	c6in.8xlarge	emr-5.36.1, emr-6.10.0
	c6in.12xlarge	emr-5.36.1, emr-6.10.0
	c6in.16xlarge	emr-5.36.1, emr-6.10.0

Kelas Instans	Tipe instans	Versi EMR Amazon yang didukung minimum (5.x, 6.x, 7.x)
	c6in.24xlarge	emr-5.36.1, emr-6.10.0
	c6in.32xlarge	emr-5.36.1, emr-6.10.0
Komputasi yang Dipercepat	g4dn.xlarge	emr-5.30.0, emr-6.0.0, emr-7.0.0
	g4dn.2xlarge	emr-5.30.0, emr-6.0.0, emr-7.0.0
	g4dn.4xlarge	emr-5.30.0, emr-6.0.0, emr-7.0.0
	g4dn.8xlarge	emr-5.30.0, emr-6.0.0, emr-7.0.0
	g4dn.12xlarge	emr-5.30.0, emr-6.0.0, emr-7.0.0
	g4dn.16xlarge	emr-5.30.0, emr-6.0.0, emr-7.0.0
Memori Dioptimalkan	r5.xlarge	emr-5.29.0, emr-6.0.0, emr-7.0.0
	r5.2xlarge	emr-5.29.0, emr-6.0.0, emr-7.0.0
	r5.4xlarge	emr-5.29.0, emr-6.0.0, emr-7.0.0
	r5.8xlarge	emr-5.29.0, emr-6.0.0, emr-7.0.0
	r5.12xlarge	emr-5.29.0, emr-6.0.0, emr-7.0.0

Kelas Instans	Tipe instans	Versi EMR Amazon yang didukung minimum (5.x, 6.x, 7.x)
	r5.16xlarge	emr-5.29.0, emr-6.0.0, emr-7.0.0
	r5.24xlarge	emr-5.29.0, emr-6.0.0, emr-7.0.0
	r5a.xlarge	emr-5.29.0, emr-6.0.0, emr-7.0.0
	r5a.2xlarge	emr-5.29.0, emr-6.0.0, emr-7.0.0
	r5a.4xlarge	emr-5.29.0, emr-6.0.0, emr-7.0.0
	r5a.8xlarge	emr-5.29.0, emr-6.0.0, emr-7.0.0
	r5a.12xlarge	emr-5.29.0, emr-6.0.0, emr-7.0.0
	r5a.16xlarge	emr-5.29.0, emr-6.0.0, emr-7.0.0
	r5a.24xlarge	emr-5.29.0, emr-6.0.0, emr-7.0.0
	r5b.xlarge	emr-5.33.0, emr-6.3.0, emr-7.0.0
	r5b.2xlarge	emr-5.33.0, emr-6.3.0, emr-7.0.0
	r5b.4xlarge	emr-5.33.0, emr-6.3.0, emr-7.0.0

Kelas Instans	Tipe instans	Versi EMR Amazon yang didukung minimum (5.x, 6.x, 7.x)
	r5b.8xlarge	emr-5.33.0, emr-6.3.0, emr-7.0.0
	r5b.12xlarge	emr-5.33.0, emr-6.3.0, emr-7.0.0
	r5b.16xlarge	emr-5.33.0, emr-6.3.0, emr-7.0.0
	r5b.24xlarge	emr-5.33.0, emr-6.3.0, emr-7.0.0
	r5d.xlarge	emr-5.29.0, emr-6.0.0, emr-7.0.0
	r5d.2xlarge	emr-5.29.0, emr-6.0.0, emr-7.0.0
	r5d.4xlarge	emr-5.29.0, emr-6.0.0, emr-7.0.0
	r5d.8xlarge	emr-5.29.0, emr-6.0.0, emr-7.0.0
	r5d.12xlarge	emr-5.29.0, emr-6.0.0, emr-7.0.0
	r5d.16xlarge	emr-5.29.0, emr-6.0.0, emr-7.0.0
	r5d.24xlarge	emr-5.29.0, emr-6.0.0, emr-7.0.0
	r5dn.xlarge	emr-5.31.0, emr-6.1.0, emr-7.0.0

Kelas Instans	Tipe instans	Versi EMR Amazon yang didukung minimum (5.x, 6.x, 7.x)
	r5dn.2xlarge	emr-5.31.0, emr-6.1.0, emr-7.0.0
	r5dn.4xlarge	emr-5.31.0, emr-6.1.0, emr-7.0.0
	r5dn.8xlarge	emr-5.31.0, emr-6.1.0, emr-7.0.0
	r5dn.12xlarge	emr-5.31.0, emr-6.1.0, emr-7.0.0
	r5dn.16xlarge	emr-5.31.0, emr-6.1.0, emr-7.0.0
	r5dn.24xlarge	emr-5.31.0, emr-6.1.0, emr-7.0.0
	r5n.xlarge	emr-5.31.0, emr-6.1.0, emr-7.0.0
	r5n.2xlarge	emr-5.31.0, emr-6.1.0, emr-7.0.0
	r5n.4xlarge	emr-5.31.0, emr-6.1.0, emr-7.0.0
	r5n.8xlarge	emr-5.31.0, emr-6.1.0, emr-7.0.0
	r5n.12xlarge	emr-5.31.0, emr-6.1.0, emr-7.0.0
	r5n.16xlarge	emr-5.31.0, emr-6.1.0, emr-7.0.0

Kelas Instans	Tipe instans	Versi EMR Amazon yang didukung minimum (5.x, 6.x, 7.x)
	r5n.24xlarge	emr-5.31.0, emr-6.1.0, emr-7.0.0
	r6g.xlarge	emr-5.31.0, emr-6.1.0, emr-7.0.0
	r6g.2xlarge	emr-5.31.0, emr-6.1.0, emr-7.0.0
	r6g.4xlarge	emr-5.31.0, emr-6.1.0, emr-7.0.0
	r6g.8xlarge	emr-5.31.0, emr-6.1.0, emr-7.0.0
	r6g.12xlarge	emr-5.31.0, emr-6.1.0, emr-7.0.0
	r6g.16xlarge	emr-5.31.0, emr-6.1.0, emr-7.0.0
	r6i.xlarge	emr-5.35.0, emr-6.6.0
	r6i.2xlarge	emr-5.35.0, emr-6.6.0
	r6i.4xlarge	emr-5.35.0, emr-6.6.0
	r6i.8xlarge	emr-5.35.0, emr-6.6.0
	r6i.12xlarge	emr-5.35.0, emr-6.6.0
	r6i.16xlarge	emr-5.35.0, emr-6.6.0
	r6i.24xlarge	emr-5.35.0, emr-6.6.0
	r6i.32xlarge	emr-5.35.0, emr-6.6.0

Kelas Instans	Tipe instans	Versi EMR Amazon yang didukung minimum (5.x, 6.x, 7.x)
	x2idn.16xlarge	emr-5.36.1, emr-6.9.0
	x2idn.24xlarge	emr-5.36.1, emr-6.9.0
	x2idn.32xlarge	emr-5.36.1, emr-6.9.0
	x2iedn.xlarge	emr-5.36.1, emr-6.9.0
	x2iedn.2xlarge	emr-5.36.1, emr-6.9.0
	x2iedn.4xlarge	emr-5.36.1, emr-6.9.0
	x2iedn.8xlarge	emr-5.36.1, emr-6.9.0
	x2iedn.16xlarge	emr-5.36.1, emr-6.9.0
	x2iedn.24xlarge	emr-5.36.1, emr-6.9.0
	x2iedn.32xlarge	emr-5.36.1, emr-6.9.0
Penyimpanan Dioptimalkan	i3.xlarge	emr-5.29.0, emr-6.0.0, emr-7.0.0
	i3.2xlarge	emr-5.29.0, emr-6.0.0, emr-7.0.0
	i3.4xlarge	emr-5.29.0, emr-6.0.0, emr-7.0.0
	i3.8xlarge	emr-5.29.0, emr-6.0.0, emr-7.0.0
	i3.16xlarge	emr-5.29.0, emr-6.0.0, emr-7.0.0
	i3en.xlarge	emr-5.29.0, emr-6.0.0, emr-7.0.0

Kelas Instans	Tipe instans	Versi EMR Amazon yang didukung minimum (5.x, 6.x, 7.x)
	i3en.2xlarge	emr-5.29.0, emr-6.0.0, emr-7.0.0
	i3en.3xlarge	emr-5.29.0, emr-6.0.0, emr-7.0.0
	i3en.6xlarge	emr-5.29.0, emr-6.0.0, emr-7.0.0
	i3en.12xlarge	emr-5.29.0, emr-6.0.0, emr-7.0.0
	i3en.24xlarge	emr-5.29.0, emr-6.0.0, emr-7.0.0
	i4i.xlarge	emr-5.36.1, emr-6.8.0
	i4i.2xlarge	emr-5.36.1, emr-6.8.0
	i4i.4xlarge	emr-5.36.1, emr-6.8.0
	i4i.8xlarge	emr-5.36.1, emr-6.8.0
	i4i.12xlarge	emr-5.36.1, emr-6.10.0
	i4i.16xlarge	emr-5.36.1, emr-6.8.0
	i4i.24xlarge	emr-5.36.1, emr-6.10.0
	i4i.32xlarge	emr-5.36.1, emr-6.8.0

Eropa (Spanyol) - eu-selatan-2

Kelas Instans	Tipe instans	Versi EMR Amazon yang didukung minimum (5.x, 6.x, 7.x)
Tujuan Umum	m5.xlarge	emr-5.36.0, emr-6.7.0
	m5.2xlarge	emr-5.36.0, emr-6.7.0
	m5.4xlarge	emr-5.36.0, emr-6.7.0
	m5.8xlarge	emr-5.36.0, emr-6.7.0
	m5.12xlarge	emr-5.36.0, emr-6.7.0
	m5.16xlarge	emr-5.36.0, emr-6.7.0
	m5.24xlarge	emr-5.36.0, emr-6.7.0
	m5d.xlarge	emr-5.36.0, emr-6.7.0
	m5d.2xlarge	emr-5.36.0, emr-6.7.0
	m5d.4xlarge	emr-5.36.0, emr-6.7.0
	m5d.8xlarge	emr-5.36.0, emr-6.7.0
	m5d.12xlarge	emr-5.36.0, emr-6.7.0
	m5d.16xlarge	emr-5.36.0, emr-6.7.0
	m5d.24xlarge	emr-5.36.0, emr-6.7.0
	m6g.xlarge	emr-5.36.0, emr-6.7.0
	m6g.2xlarge	emr-5.36.0, emr-6.7.0
	m6g.4xlarge	emr-5.36.0, emr-6.7.0
	m6g.8xlarge	emr-5.36.0, emr-6.7.0

Kelas Instans	Tipe instans	Versi EMR Amazon yang didukung minimum (5.x, 6.x, 7.x)
	m6g.12xlarge	emr-5.36.0, emr-6.7.0
	m6g.16xlarge	emr-5.36.0, emr-6.7.0
	m6gd.xlarge	emr-5.36.0, emr-6.7.0
	m6gd.2xlarge	emr-5.36.0, emr-6.7.0
	m6gd.4xlarge	emr-5.36.0, emr-6.7.0
	m6gd.8xlarge	emr-5.36.0, emr-6.7.0
	m6gd.12xlarge	emr-5.36.0, emr-6.7.0
	m6gd.16xlarge	emr-5.36.0, emr-6.7.0
	m7g.xlarge	emr-5.36.1, emr-6.10.0
	m7g.2xlarge	emr-5.36.1, emr-6.10.0
	m7g.4xlarge	emr-5.36.1, emr-6.10.0
	m7g.8xlarge	emr-5.36.1, emr-6.10.0
	m7g.12xlarge	emr-5.36.1, emr-6.10.0
	m7g.16xlarge	emr-5.36.1, emr-6.10.0
	m7i.xlarge	emr-5.36.1, emr-6.10.0
	m7i.2xlarge	emr-5.36.1, emr-6.10.0
	m7i.4xlarge	emr-5.36.1, emr-6.10.0
	m7i.8xlarge	emr-5.36.1, emr-6.10.0
	m7i.12xlarge	emr-5.36.1, emr-6.10.0

Kelas Instans	Tipe instans	Versi EMR Amazon yang didukung minimum (5.x, 6.x, 7.x)
	m7i.16xlarge	emr-5.36.1, emr-6.10.0
	m7i.24xlarge	emr-5.36.1, emr-6.10.0
	m7i.48xlarge	emr-5.36.1, emr-6.10.0
	m7i-flex.xlarge	emr-5.36.1, emr-6.10.0
	m7i-flex.2xlarge	emr-5.36.1, emr-6.10.0
	m7i-flex.4xlarge	emr-5.36.1, emr-6.10.0
	m7i-flex.8xlarge	emr-5.36.1, emr-6.10.0
Komputasi Dioptimalkan	c5.xlarge	emr-5.36.0, emr-6.7.0
	c5.2xlarge	emr-5.36.0, emr-6.7.0
	c5.4xlarge	emr-5.36.0, emr-6.7.0
	c5.9xlarge	emr-5.36.0, emr-6.7.0
	c5.12xlarge	emr-5.36.0, emr-6.7.0
	c5.18xlarge	emr-5.36.0, emr-6.7.0
	c5.24xlarge	emr-5.36.0, emr-6.7.0
	c5d.xlarge	emr-5.36.0, emr-6.7.0
	c5d.2xlarge	emr-5.36.0, emr-6.7.0
	c5d.4xlarge	emr-5.36.0, emr-6.7.0
	c5d.9xlarge	emr-5.36.0, emr-6.7.0
	c5d.12xlarge	emr-5.36.0, emr-6.7.0

Kelas Instans	Tipe instans	Versi EMR Amazon yang didukung minimum (5.x, 6.x, 7.x)
	c5d.18xlarge	emr-5.36.0, emr-6.7.0
	c5d.24xlarge	emr-5.36.0, emr-6.7.0
	c6g.xlarge	emr-5.36.0, emr-6.7.0
	c6g.2xlarge	emr-5.36.0, emr-6.7.0
	c6g.4xlarge	emr-5.36.0, emr-6.7.0
	c6g.8xlarge	emr-5.36.0, emr-6.7.0
	c6g.12xlarge	emr-5.36.0, emr-6.7.0
	c6g.16xlarge	emr-5.36.0, emr-6.7.0
	c7g.xlarge	emr-5.36.1, emr-6.7.0
	c7g.2xlarge	emr-5.36.1, emr-6.7.0
	c7g.4xlarge	emr-5.36.1, emr-6.7.0
	c7g.8xlarge	emr-5.36.1, emr-6.7.0
	c7g.12xlarge	emr-5.36.1, emr-6.7.0
	c7g.16xlarge	emr-5.36.1, emr-6.7.0
	c7i.xlarge	emr-5.36.1, emr-6.10.0
	c7i.2xlarge	emr-5.36.1, emr-6.10.0
	c7i.4xlarge	emr-5.36.1, emr-6.10.0
	c7i.8xlarge	emr-5.36.1, emr-6.10.0
	c7i.12xlarge	emr-5.36.1, emr-6.10.0

Kelas Instans	Tipe instans	Versi EMR Amazon yang didukung minimum (5.x, 6.x, 7.x)
	c7i.16xlarge	emr-5.36.1, emr-6.10.0
	c7i.24xlarge	emr-5.36.1, emr-6.10.0
	c7i.48xlarge	emr-5.36.1, emr-6.10.0
Memori Dioptimalkan	r5.xlarge	emr-5.36.0, emr-6.7.0
	r5.2xlarge	emr-5.36.0, emr-6.7.0
	r5.4xlarge	emr-5.36.0, emr-6.7.0
	r5.8xlarge	emr-5.36.0, emr-6.7.0
	r5.12xlarge	emr-5.36.0, emr-6.7.0
	r5.16xlarge	emr-5.36.0, emr-6.7.0
	r5.24xlarge	emr-5.36.0, emr-6.7.0
	r5d.xlarge	emr-5.36.0, emr-6.7.0
	r5d.2xlarge	emr-5.36.0, emr-6.7.0
	r5d.4xlarge	emr-5.36.0, emr-6.7.0
	r5d.8xlarge	emr-5.36.0, emr-6.7.0
	r5d.12xlarge	emr-5.36.0, emr-6.7.0
	r5d.16xlarge	emr-5.36.0, emr-6.7.0
	r5d.24xlarge	emr-5.36.0, emr-6.7.0
	r6g.xlarge	emr-5.36.0, emr-6.7.0
	r6g.2xlarge	emr-5.36.0, emr-6.7.0

Kelas Instans	Tipe instans	Versi EMR Amazon yang didukung minimum (5.x, 6.x, 7.x)
	r6g.4xlarge	emr-5.36.0, emr-6.7.0
	r6g.8xlarge	emr-5.36.0, emr-6.7.0
	r6g.12xlarge	emr-5.36.0, emr-6.7.0
	r6g.16xlarge	emr-5.36.0, emr-6.7.0
	r7g.xlarge	emr-5.36.1, emr-6.10.0
	r7g.2xlarge	emr-5.36.1, emr-6.10.0
	r7g.4xlarge	emr-5.36.1, emr-6.10.0
	r7g.8xlarge	emr-5.36.1, emr-6.10.0
	r7g.12xlarge	emr-5.36.1, emr-6.10.0
	r7g.16xlarge	emr-5.36.1, emr-6.10.0
	r7i.xlarge	emr-5.36.1, emr-6.10.0
	r7i.2xlarge	emr-5.36.1, emr-6.10.0
	r7i.4xlarge	emr-5.36.1, emr-6.10.0
	r7i.8xlarge	emr-5.36.1, emr-6.10.0
	r7i.12xlarge	emr-5.36.1, emr-6.10.0
	r7i.16xlarge	emr-5.36.1, emr-6.10.0
	r7i.24xlarge	emr-5.36.1, emr-6.10.0
	r7i.48xlarge	emr-5.36.1, emr-6.10.0
	x2idn.16xlarge	emr-5.36.1, emr-6.9.0

Kelas Instans	Tipe instans	Versi EMR Amazon yang didukung minimum (5.x, 6.x, 7.x)
	x2idn.24xlarge	emr-5.36.1, emr-6.9.0
	x2idn.32xlarge	emr-5.36.1, emr-6.9.0
	x2iedn.xlarge	emr-5.36.1, emr-6.9.0
	x2iedn.2xlarge	emr-5.36.1, emr-6.9.0
	x2iedn.4xlarge	emr-5.36.1, emr-6.9.0
	x2iedn.8xlarge	emr-5.36.1, emr-6.9.0
	x2iedn.16xlarge	emr-5.36.1, emr-6.9.0
	x2iedn.24xlarge	emr-5.36.1, emr-6.9.0
	x2iedn.32xlarge	emr-5.36.1, emr-6.9.0
Penyimpanan Dioptimalkan	i3.xlarge	emr-5.36.0, emr-6.7.0
	i3.2xlarge	emr-5.36.0, emr-6.7.0
	i3.4xlarge	emr-5.36.0, emr-6.7.0
	i3.8xlarge	emr-5.36.0, emr-6.7.0
	i3.16xlarge	emr-5.36.0, emr-6.7.0
	i3en.xlarge	emr-5.36.0, emr-6.7.0
	i3en.2xlarge	emr-5.36.0, emr-6.7.0
	i3en.3xlarge	emr-5.36.0, emr-6.7.0
	i3en.6xlarge	emr-5.36.0, emr-6.7.0
	i3en.12xlarge	emr-5.36.0, emr-6.7.0

Kelas Instans	Tipe instans	Versi EMR Amazon yang didukung minimum (5.x, 6.x, 7.x)
	i3en.24xlarge	emr-5.36.0, emr-6.7.0

Europe (Paris) - eu-west-3

Kelas Instans	Tipe instans	Versi EMR Amazon yang didukung minimum (5.x, 6.x, 7.x)
Tujuan Umum	m5.xlarge	emr-5.13.0, emr-6.0.0, emr-7.0.0
	m5.2xlarge	emr-5.13.0, emr-6.0.0, emr-7.0.0
	m5.4xlarge	emr-5.13.0, emr-6.0.0, emr-7.0.0
	m5.8xlarge	emr-5.13.0, emr-6.0.0, emr-7.0.0
	m5.12xlarge	emr-5.13.0, emr-6.0.0, emr-7.0.0
	m5.16xlarge	emr-5.13.0, emr-6.0.0, emr-7.0.0
	m5.24xlarge	emr-5.13.0, emr-6.0.0, emr-7.0.0
	m5a.xlarge	emr-5.20.0, emr-6.0.0, emr-7.0.0
	m5a.2xlarge	emr-5.20.0, emr-6.0.0, emr-7.0.0

Kelas Instans	Tipe instans	Versi EMR Amazon yang didukung minimum (5.x, 6.x, 7.x)
	m5a.4xlarge	emr-5.20.0, emr-6.0.0, emr-7.0.0
	m5a.8xlarge	emr-5.20.0, emr-6.0.0, emr-7.0.0
	m5a.12xlarge	emr-5.20.0, emr-6.0.0, emr-7.0.0
	m5a.16xlarge	emr-5.20.0, emr-6.0.0, emr-7.0.0
	m5a.24xlarge	emr-5.20.0, emr-6.0.0, emr-7.0.0
	m5ad.xlarge	emr-5.20.0, emr-6.0.0, emr-7.0.0
	m5ad.2xlarge	emr-5.20.0, emr-6.0.0, emr-7.0.0
	m5ad.4xlarge	emr-5.20.0, emr-6.0.0, emr-7.0.0
	m5ad.8xlarge	emr-5.33.0, emr-6.3.0, emr-7.0.0
	m5ad.12xlarge	emr-5.20.0, emr-6.0.0, emr-7.0.0
	m5ad.16xlarge	emr-5.33.0, emr-6.3.0, emr-7.0.0
	m5ad.24xlarge	emr-5.20.0, emr-6.0.0, emr-7.0.0

Kelas Instans	Tipe instans	Versi EMR Amazon yang didukung minimum (5.x, 6.x, 7.x)
	m5d.xlarge	emr-5.13.0, emr-6.0.0, emr-7.0.0
	m5d.2xlarge	emr-5.13.0, emr-6.0.0, emr-7.0.0
	m5d.4xlarge	emr-5.13.0, emr-6.0.0, emr-7.0.0
	m5d.8xlarge	emr-5.13.0, emr-6.0.0, emr-7.0.0
	m5d.12xlarge	emr-5.13.0, emr-6.0.0, emr-7.0.0
	m5d.16xlarge	emr-5.13.0, emr-6.0.0, emr-7.0.0
	m5d.24xlarge	emr-5.13.0, emr-6.0.0, emr-7.0.0
	m6g.xlarge	emr-5.30.0, emr-6.1.0, emr-7.0.0
	m6g.2xlarge	emr-5.30.0, emr-6.1.0, emr-7.0.0
	m6g.4xlarge	emr-5.30.0, emr-6.1.0, emr-7.0.0
	m6g.8xlarge	emr-5.30.0, emr-6.1.0, emr-7.0.0
	m6g.12xlarge	emr-5.30.0, emr-6.1.0, emr-7.0.0

Kelas Instans	Tipe instans	Versi EMR Amazon yang didukung minimum (5.x, 6.x, 7.x)
	m6g.16xlarge	emr-5.30.0, emr-6.1.0, emr-7.0.0
	m6gd.xlarge	emr-5.33.0, emr-6.3.0, emr-7.0.0
	m6gd.2xlarge	emr-5.33.0, emr-6.3.0, emr-7.0.0
	m6gd.4xlarge	emr-5.33.0, emr-6.3.0, emr-7.0.0
	m6gd.8xlarge	emr-5.33.0, emr-6.3.0, emr-7.0.0
	m6gd.12xlarge	emr-5.33.0, emr-6.3.0, emr-7.0.0
	m6gd.16xlarge	emr-5.33.0, emr-6.3.0, emr-7.0.0
	m6i.xlarge	emr-5.35.0, emr-6.6.0
	m6i.2xlarge	emr-5.35.0, emr-6.6.0
	m6i.4xlarge	emr-5.35.0, emr-6.6.0
	m6i.8xlarge	emr-5.35.0, emr-6.6.0
	m6i.12xlarge	emr-5.35.0, emr-6.6.0
	m6i.16xlarge	emr-5.35.0, emr-6.6.0
	m6i.24xlarge	emr-5.35.0, emr-6.6.0
	m6i.32xlarge	emr-5.35.0, emr-6.6.0

Kelas Instans	Tipe instans	Versi EMR Amazon yang didukung minimum (5.x, 6.x, 7.x)
Komputasi Dioptimalkan	c5.xlarge	emr-5.13.0, emr-6.0.0, emr-7.0.0
	c5.2xlarge	emr-5.13.0, emr-6.0.0, emr-7.0.0
	c5.4xlarge	emr-5.13.0, emr-6.0.0, emr-7.0.0
	c5.9xlarge	emr-5.13.0, emr-6.0.0, emr-7.0.0
	c5.12xlarge	emr-5.13.0, emr-6.0.0, emr-7.0.0
	c5.18xlarge	emr-5.13.0, emr-6.0.0, emr-7.0.0
	c5.24xlarge	emr-5.13.0, emr-6.0.0, emr-7.0.0
	c5a.xlarge	emr-5.31.0, emr-6.1.0, emr-7.0.0
	c5a.2xlarge	emr-5.31.0, emr-6.1.0, emr-7.0.0
	c5a.4xlarge	emr-5.31.0, emr-6.1.0, emr-7.0.0
	c5a.8xlarge	emr-5.31.0, emr-6.1.0, emr-7.0.0
	c5a.12xlarge	emr-5.31.0, emr-6.1.0, emr-7.0.0

Kelas Instans	Tipe instans	Versi EMR Amazon yang didukung minimum (5.x, 6.x, 7.x)
	c5a.16xlarge	emr-5.31.0, emr-6.1.0, emr-7.0.0
	c5a.24xlarge	emr-5.31.0, emr-6.1.0, emr-7.0.0
	c5d.xlarge	emr-5.13.0, emr-6.0.0, emr-7.0.0
	c5d.2xlarge	emr-5.13.0, emr-6.0.0, emr-7.0.0
	c5d.4xlarge	emr-5.13.0, emr-6.0.0, emr-7.0.0
	c5d.9xlarge	emr-5.13.0, emr-6.0.0, emr-7.0.0
	c5d.18xlarge	emr-5.13.0, emr-6.0.0, emr-7.0.0
	c5n.xlarge	emr-5.20.0, emr-6.0.0, emr-7.0.0
	c5n.2xlarge	emr-5.20.0, emr-6.0.0, emr-7.0.0
	c5n.4xlarge	emr-5.20.0, emr-6.0.0, emr-7.0.0
	c5n.9xlarge	emr-5.20.0, emr-6.0.0, emr-7.0.0
	c5n.18xlarge	emr-5.20.0, emr-6.0.0, emr-7.0.0

Kelas Instans	Tipe instans	Versi EMR Amazon yang didukung minimum (5.x, 6.x, 7.x)
	c6g.xlarge	emr-5.31.0, emr-6.1.0, emr-7.0.0
	c6g.2xlarge	emr-5.31.0, emr-6.1.0, emr-7.0.0
	c6g.4xlarge	emr-5.31.0, emr-6.1.0, emr-7.0.0
	c6g.8xlarge	emr-5.31.0, emr-6.1.0, emr-7.0.0
	c6g.12xlarge	emr-5.31.0, emr-6.1.0, emr-7.0.0
	c6g.16xlarge	emr-5.31.0, emr-6.1.0, emr-7.0.0
	c6gd.xlarge	emr-5.33.0, emr-6.3.0, emr-7.0.0
	c6gd.2xlarge	emr-5.33.0, emr-6.3.0, emr-7.0.0
	c6gd.4xlarge	emr-5.33.0, emr-6.3.0, emr-7.0.0
	c6gd.8xlarge	emr-5.33.0, emr-6.3.0, emr-7.0.0
	c6gd.12xlarge	emr-5.33.0, emr-6.3.0, emr-7.0.0
	c6gd.16xlarge	emr-5.33.0, emr-6.3.0, emr-7.0.0

Kelas Instans	Tipe instans	Versi EMR Amazon yang didukung minimum (5.x, 6.x, 7.x)
	c6gn.xlarge	emr-5.33.0, emr-6.3.0, emr-7.0.0
	c6gn.2xlarge	emr-5.33.0, emr-6.3.0, emr-7.0.0
	c6gn.4xlarge	emr-5.33.0, emr-6.3.0, emr-7.0.0
	c6gn.8xlarge	emr-5.33.0, emr-6.3.0, emr-7.0.0
	c6gn.12xlarge	emr-5.33.0, emr-6.3.0, emr-7.0.0
	c6gn.16xlarge	emr-5.33.0, emr-6.3.0, emr-7.0.0
	c6i.xlarge	emr-5.35.0, emr-6.6.0
	c6i.2xlarge	emr-5.35.0, emr-6.6.0
	c6i.4xlarge	emr-5.35.0, emr-6.6.0
	c6i.8xlarge	emr-5.35.0, emr-6.6.0
	c6i.12xlarge	emr-5.35.0, emr-6.6.0
	c6i.16xlarge	emr-5.35.0, emr-6.6.0
	c6i.24xlarge	emr-5.35.0, emr-6.6.0
	c6i.32xlarge	emr-5.35.0, emr-6.6.0
	c6in.xlarge	emr-5.36.1, emr-6.10.0
	c6in.2xlarge	emr-5.36.1, emr-6.10.0

Kelas Instans	Tipe instans	Versi EMR Amazon yang didukung minimum (5.x, 6.x, 7.x)
	c6in.4xlarge	emr-5.36.1, emr-6.10.0
	c6in.8xlarge	emr-5.36.1, emr-6.10.0
	c6in.12xlarge	emr-5.36.1, emr-6.10.0
	c6in.16xlarge	emr-5.36.1, emr-6.10.0
	c6in.24xlarge	emr-5.36.1, emr-6.10.0
	c6in.32xlarge	emr-5.36.1, emr-6.10.0
Komputasi yang Dipercepat	g4dn.xlarge	emr-5.30.0, emr-6.0.0, emr-7.0.0
	g4dn.2xlarge	emr-5.30.0, emr-6.0.0, emr-7.0.0
	g4dn.4xlarge	emr-5.30.0, emr-6.0.0, emr-7.0.0
	g4dn.8xlarge	emr-5.30.0, emr-6.0.0, emr-7.0.0
	g4dn.12xlarge	emr-5.30.0, emr-6.0.0, emr-7.0.0
	g4dn.16xlarge	emr-5.30.0, emr-6.0.0, emr-7.0.0
Memori Dioptimalkan	r5.xlarge	emr-5.13.0, emr-6.0.0, emr-7.0.0
	r5.2xlarge	emr-5.13.0, emr-6.0.0, emr-7.0.0

Kelas Instans	Tipe instans	Versi EMR Amazon yang didukung minimum (5.x, 6.x, 7.x)
	r5.4xlarge	emr-5.13.0, emr-6.0.0, emr-7.0.0
	r5.8xlarge	emr-5.13.0, emr-6.0.0, emr-7.0.0
	r5.12xlarge	emr-5.13.0, emr-6.0.0, emr-7.0.0
	r5.16xlarge	emr-5.13.0, emr-6.0.0, emr-7.0.0
	r5.24xlarge	emr-5.13.0, emr-6.0.0, emr-7.0.0
	r5a.xlarge	emr-5.20.0, emr-6.0.0, emr-7.0.0
	r5a.2xlarge	emr-5.20.0, emr-6.0.0, emr-7.0.0
	r5a.4xlarge	emr-5.20.0, emr-6.0.0, emr-7.0.0
	r5a.8xlarge	emr-5.20.0, emr-6.0.0, emr-7.0.0
	r5a.12xlarge	emr-5.20.0, emr-6.0.0, emr-7.0.0
	r5a.16xlarge	emr-5.20.0, emr-6.0.0, emr-7.0.0
	r5a.24xlarge	emr-5.20.0, emr-6.0.0, emr-7.0.0

Kelas Instans	Tipe instans	Versi EMR Amazon yang didukung minimum (5.x, 6.x, 7.x)
	r5ad.xlarge	emr-5.20.0, emr-6.0.0, emr-7.0.0
	r5ad.2xlarge	emr-5.20.0, emr-6.0.0, emr-7.0.0
	r5ad.4xlarge	emr-5.20.0, emr-6.0.0, emr-7.0.0
	r5ad.8xlarge	emr-5.33.0, emr-6.3.0, emr-7.0.0
	r5ad.12xlarge	emr-5.20.0, emr-6.0.0, emr-7.0.0
	r5ad.16xlarge	emr-5.33.0, emr-6.3.0, emr-7.0.0
	r5ad.24xlarge	emr-5.20.0, emr-6.0.0, emr-7.0.0
	r5d.xlarge	emr-5.13.0, emr-6.0.0, emr-7.0.0
	r5d.2xlarge	emr-5.13.0, emr-6.0.0, emr-7.0.0
	r5d.4xlarge	emr-5.13.0, emr-6.0.0, emr-7.0.0
	r5d.8xlarge	emr-5.13.0, emr-6.0.0, emr-7.0.0
	r5d.12xlarge	emr-5.13.0, emr-6.0.0, emr-7.0.0

Kelas Instans	Tipe instans	Versi EMR Amazon yang didukung minimum (5.x, 6.x, 7.x)
	r5d.16xlarge	emr-5.13.0, emr-6.0.0, emr-7.0.0
	r5d.24xlarge	emr-5.13.0, emr-6.0.0, emr-7.0.0
	r5dn.xlarge	emr-5.31.0, emr-6.1.0, emr-7.0.0
	r5dn.2xlarge	emr-5.31.0, emr-6.1.0, emr-7.0.0
	r5dn.4xlarge	emr-5.31.0, emr-6.1.0, emr-7.0.0
	r5dn.8xlarge	emr-5.31.0, emr-6.1.0, emr-7.0.0
	r5dn.12xlarge	emr-5.31.0, emr-6.1.0, emr-7.0.0
	r5dn.16xlarge	emr-5.31.0, emr-6.1.0, emr-7.0.0
	r5dn.24xlarge	emr-5.31.0, emr-6.1.0, emr-7.0.0
	r5n.xlarge	emr-5.31.0, emr-6.1.0, emr-7.0.0
	r5n.2xlarge	emr-5.31.0, emr-6.1.0, emr-7.0.0
	r5n.4xlarge	emr-5.31.0, emr-6.1.0, emr-7.0.0

Kelas Instans	Tipe instans	Versi EMR Amazon yang didukung minimum (5.x, 6.x, 7.x)
	r5n.8xlarge	emr-5.31.0, emr-6.1.0, emr-7.0.0
	r5n.12xlarge	emr-5.31.0, emr-6.1.0, emr-7.0.0
	r5n.16xlarge	emr-5.31.0, emr-6.1.0, emr-7.0.0
	r5n.24xlarge	emr-5.31.0, emr-6.1.0, emr-7.0.0
	r6g.xlarge	emr-5.31.0, emr-6.1.0, emr-7.0.0
	r6g.2xlarge	emr-5.31.0, emr-6.1.0, emr-7.0.0
	r6g.4xlarge	emr-5.31.0, emr-6.1.0, emr-7.0.0
	r6g.8xlarge	emr-5.31.0, emr-6.1.0, emr-7.0.0
	r6g.12xlarge	emr-5.31.0, emr-6.1.0, emr-7.0.0
	r6g.16xlarge	emr-5.31.0, emr-6.1.0, emr-7.0.0
	r6gd.xlarge	emr-5.33.0, emr-6.3.0, emr-7.0.0
	r6gd.2xlarge	emr-5.33.0, emr-6.3.0, emr-7.0.0

Kelas Instans	Tipe instans	Versi EMR Amazon yang didukung minimum (5.x, 6.x, 7.x)
	r6gd.4xlarge	emr-5.33.0, emr-6.3.0, emr-7.0.0
	r6gd.8xlarge	emr-5.33.0, emr-6.3.0, emr-7.0.0
	r6gd.12xlarge	emr-5.33.0, emr-6.3.0, emr-7.0.0
	r6gd.16xlarge	emr-5.33.0, emr-6.3.0, emr-7.0.0
	r6i.xlarge	emr-5.35.0, emr-6.6.0
	r6i.2xlarge	emr-5.35.0, emr-6.6.0
	r6i.4xlarge	emr-5.35.0, emr-6.6.0
	r6i.8xlarge	emr-5.35.0, emr-6.6.0
	r6i.12xlarge	emr-5.35.0, emr-6.6.0
	r6i.16xlarge	emr-5.35.0, emr-6.6.0
	r6i.24xlarge	emr-5.35.0, emr-6.6.0
	r6i.32xlarge	emr-5.35.0, emr-6.6.0
	x1.16xlarge	emr-5.36.1, emr-6.9.0, emr-7.0.0
	x1.32xlarge	emr-5.36.1, emr-6.9.0, emr-7.0.0
	x2idn.16xlarge	emr-5.36.1, emr-6.9.0
	x2idn.24xlarge	emr-5.36.1, emr-6.9.0

Kelas Instans	Tipe instans	Versi EMR Amazon yang didukung minimum (5.x, 6.x, 7.x)
	x2idn.32xlarge	emr-5.36.1, emr-6.9.0
	x2iedn.xlarge	emr-5.36.1, emr-6.9.0
	x2iedn.2xlarge	emr-5.36.1, emr-6.9.0
	x2iedn.4xlarge	emr-5.36.1, emr-6.9.0
	x2iedn.8xlarge	emr-5.36.1, emr-6.9.0
	x2iedn.16xlarge	emr-5.36.1, emr-6.9.0
	x2iedn.24xlarge	emr-5.36.1, emr-6.9.0
	x2iedn.32xlarge	emr-5.36.1, emr-6.9.0
Penyimpanan Dioptimalkan	i3.xlarge	emr-5.9.0, emr-6.0.0, emr-7.0.0
	i3.2xlarge	emr-5.9.0, emr-6.0.0, emr-7.0.0
	i3.4xlarge	emr-5.9.0, emr-6.0.0, emr-7.0.0
	i3.8xlarge	emr-5.9.0, emr-6.0.0, emr-7.0.0
	i3.16xlarge	emr-5.9.0, emr-6.0.0, emr-7.0.0
	i3en.xlarge	emr-5.25.0, emr-6.0.0, emr-7.0.0
	i3en.2xlarge	emr-5.25.0, emr-6.0.0, emr-7.0.0

Kelas Instans	Tipe instans	Versi EMR Amazon yang didukung minimum (5.x, 6.x, 7.x)
	i3en.3xlarge	emr-5.25.0, emr-6.0.0, emr-7.0.0
	i3en.6xlarge	emr-5.25.0, emr-6.0.0, emr-7.0.0
	i3en.12xlarge	emr-5.25.0, emr-6.0.0, emr-7.0.0
	i3en.24xlarge	emr-5.25.0, emr-6.0.0, emr-7.0.0
	i4i.xlarge	emr-5.36.1, emr-6.8.0
	i4i.2xlarge	emr-5.36.1, emr-6.8.0
	i4i.4xlarge	emr-5.36.1, emr-6.8.0
	i4i.8xlarge	emr-5.36.1, emr-6.8.0
	i4i.12xlarge	emr-5.36.1, emr-6.10.0
	i4i.16xlarge	emr-5.36.1, emr-6.8.0
	i4i.24xlarge	emr-5.36.1, emr-6.10.0
	i4i.32xlarge	emr-5.36.1, emr-6.8.0
	im4gn.xlarge	emr-5.35.0, emr-6.6.0, emr-7.0.0
	im4gn.2xlarge	emr-5.35.0, emr-6.6.0, emr-7.0.0
	im4gn.4xlarge	emr-5.35.0, emr-6.6.0, emr-7.0.0

Kelas Instans	Tipe instans	Versi EMR Amazon yang didukung minimum (5.x, 6.x, 7.x)
	im4gn.8xlarge	emr-5.35.0, emr-6.6.0, emr-7.0.0
	im4gn.16xlarge	emr-5.35.0, emr-6.6.0, emr-7.0.0
	is4gen.xlarge	emr-5.35.0, emr-6.6.0, emr-7.0.0
	is4gen.2xlarge	emr-5.35.0, emr-6.6.0, emr-7.0.0
	is4gen.4xlarge	emr-5.35.0, emr-6.6.0, emr-7.0.0
	is4gen.8xlarge	emr-5.35.0, emr-6.6.0, emr-7.0.0

Europe (Stockholm) - eu-north-1

Kelas Instans	Tipe instans	Versi EMR Amazon yang didukung minimum (5.x, 6.x, 7.x)
Tujuan Umum	m5.xlarge	emr-5.17.0, emr-6.0.0, emr-7.0.0
	m5.2xlarge	emr-5.17.0, emr-6.0.0, emr-7.0.0
	m5.4xlarge	emr-5.17.0, emr-6.0.0, emr-7.0.0

Kelas Instans	Tipe instans	Versi EMR Amazon yang didukung minimum (5.x, 6.x, 7.x)
	m5.8xlarge	emr-5.17.0, emr-6.0.0, emr-7.0.0
	m5.12xlarge	emr-5.17.0, emr-6.0.0, emr-7.0.0
	m5.16xlarge	emr-5.17.0, emr-6.0.0, emr-7.0.0
	m5.24xlarge	emr-5.17.0, emr-6.0.0, emr-7.0.0
	m5d.xlarge	emr-5.17.0, emr-6.0.0, emr-7.0.0
	m5d.2xlarge	emr-5.17.0, emr-6.0.0, emr-7.0.0
	m5d.4xlarge	emr-5.17.0, emr-6.0.0, emr-7.0.0
	m5d.8xlarge	emr-5.17.0, emr-6.0.0, emr-7.0.0
	m5d.12xlarge	emr-5.17.0, emr-6.0.0, emr-7.0.0
	m5d.16xlarge	emr-5.17.0, emr-6.0.0, emr-7.0.0
	m5d.24xlarge	emr-5.17.0, emr-6.0.0, emr-7.0.0
	m6g.xlarge	emr-5.30.0, emr-6.1.0, emr-7.0.0

Kelas Instans	Tipe instans	Versi EMR Amazon yang didukung minimum (5.x, 6.x, 7.x)
	m6g.2xlarge	emr-5.30.0, emr-6.1.0, emr-7.0.0
	m6g.4xlarge	emr-5.30.0, emr-6.1.0, emr-7.0.0
	m6g.8xlarge	emr-5.30.0, emr-6.1.0, emr-7.0.0
	m6g.12xlarge	emr-5.30.0, emr-6.1.0, emr-7.0.0
	m6g.16xlarge	emr-5.30.0, emr-6.1.0, emr-7.0.0
	m6gd.xlarge	emr-5.33.0, emr-6.3.0, emr-7.0.0
	m6gd.2xlarge	emr-5.33.0, emr-6.3.0, emr-7.0.0
	m6gd.4xlarge	emr-5.33.0, emr-6.3.0, emr-7.0.0
	m6gd.8xlarge	emr-5.33.0, emr-6.3.0, emr-7.0.0
	m6gd.12xlarge	emr-5.33.0, emr-6.3.0, emr-7.0.0
	m6gd.16xlarge	emr-5.33.0, emr-6.3.0, emr-7.0.0
	m6i.xlarge	emr-5.35.0, emr-6.6.0
	m6i.2xlarge	emr-5.35.0, emr-6.6.0

Kelas Instans	Tipe instans	Versi EMR Amazon yang didukung minimum (5.x, 6.x, 7.x)
	m6i.4xlarge	emr-5.35.0, emr-6.6.0
	m6i.8xlarge	emr-5.35.0, emr-6.6.0
	m6i.12xlarge	emr-5.35.0, emr-6.6.0
	m6i.16xlarge	emr-5.35.0, emr-6.6.0
	m6i.24xlarge	emr-5.35.0, emr-6.6.0
	m6i.32xlarge	emr-5.35.0, emr-6.6.0
	m6idn.xlarge	emr-5.36.1, emr-6.10.0
	m6idn.2xlarge	emr-5.36.1, emr-6.10.0
	m6idn.4xlarge	emr-5.36.1, emr-6.10.0
	m6idn.8xlarge	emr-5.36.1, emr-6.10.0
	m6idn.12xlarge	emr-5.36.1, emr-6.10.0
	m6idn.16xlarge	emr-5.36.1, emr-6.10.0
	m6idn.24xlarge	emr-5.36.1, emr-6.10.0
	m6idn.32xlarge	emr-5.36.1, emr-6.10.0
	m6in.xlarge	emr-5.36.1, emr-6.10.0
	m6in.2xlarge	emr-5.36.1, emr-6.10.0
	m6in.4xlarge	emr-5.36.1, emr-6.10.0
	m6in.8xlarge	emr-5.36.1, emr-6.10.0
	m6in.12xlarge	emr-5.36.1, emr-6.10.0

Kelas Instans	Tipe instans	Versi EMR Amazon yang didukung minimum (5.x, 6.x, 7.x)
	m6in.16xlarge	emr-5.36.1, emr-6.10.0
	m6in.24xlarge	emr-5.36.1, emr-6.10.0
	m6in.32xlarge	emr-5.36.1, emr-6.10.0
	m7g.xlarge	emr-5.36.1, emr-6.10.0, emr-7.0.0
	m7g.2xlarge	emr-5.36.1, emr-6.10.0, emr-7.0.0
	m7g.4xlarge	emr-5.36.1, emr-6.10.0, emr-7.0.0
	m7g.8xlarge	emr-5.36.1, emr-6.10.0, emr-7.0.0
	m7g.12xlarge	emr-5.36.1, emr-6.10.0, emr-7.0.0
	m7g.16xlarge	emr-5.36.1, emr-6.10.0, emr-7.0.0
	m7i.xlarge	emr-5.36.1, emr-6.10.0
	m7i.2xlarge	emr-5.36.1, emr-6.10.0
	m7i.4xlarge	emr-5.36.1, emr-6.10.0
	m7i.8xlarge	emr-5.36.1, emr-6.10.0
	m7i.12xlarge	emr-5.36.1, emr-6.10.0
	m7i.16xlarge	emr-5.36.1, emr-6.10.0
	m7i.24xlarge	emr-5.36.1, emr-6.10.0

Kelas Instans	Tipe instans	Versi EMR Amazon yang didukung minimum (5.x, 6.x, 7.x)
	m7i.48xlarge	emr-5.36.1, emr-6.10.0
	m7i-flex.xlarge	emr-5.36.1, emr-6.10.0
	m7i-flex.2xlarge	emr-5.36.1, emr-6.10.0
	m7i-flex.4xlarge	emr-5.36.1, emr-6.10.0
	m7i-flex.8xlarge	emr-5.36.1, emr-6.10.0
Komputasi Dioptimalkan	c5.xlarge	emr-5.17.0, emr-6.0.0, emr-7.0.0
	c5.2xlarge	emr-5.17.0, emr-6.0.0, emr-7.0.0
	c5.4xlarge	emr-5.17.0, emr-6.0.0, emr-7.0.0
	c5.9xlarge	emr-5.17.0, emr-6.0.0, emr-7.0.0
	c5.12xlarge	emr-5.17.0, emr-6.0.0, emr-7.0.0
	c5.18xlarge	emr-5.17.0, emr-6.0.0, emr-7.0.0
	c5.24xlarge	emr-5.17.0, emr-6.0.0, emr-7.0.0
	c5a.xlarge	emr-5.31.0, emr-6.1.0, emr-7.0.0
	c5a.2xlarge	emr-5.31.0, emr-6.1.0, emr-7.0.0

Kelas Instans	Tipe instans	Versi EMR Amazon yang didukung minimum (5.x, 6.x, 7.x)
	c5a.4xlarge	emr-5.31.0, emr-6.1.0, emr-7.0.0
	c5a.8xlarge	emr-5.31.0, emr-6.1.0, emr-7.0.0
	c5a.12xlarge	emr-5.31.0, emr-6.1.0, emr-7.0.0
	c5a.16xlarge	emr-5.31.0, emr-6.1.0, emr-7.0.0
	c5a.24xlarge	emr-5.31.0, emr-6.1.0, emr-7.0.0
	c5d.xlarge	emr-5.17.0, emr-6.0.0, emr-7.0.0
	c5d.2xlarge	emr-5.17.0, emr-6.0.0, emr-7.0.0
	c5d.4xlarge	emr-5.17.0, emr-6.0.0, emr-7.0.0
	c5d.9xlarge	emr-5.17.0, emr-6.0.0, emr-7.0.0
	c5d.12xlarge	emr-5.17.0, emr-6.0.0, emr-7.0.0
	c5d.18xlarge	emr-5.17.0, emr-6.0.0, emr-7.0.0
	c5d.24xlarge	emr-5.17.0, emr-6.0.0, emr-7.0.0

Kelas Instans	Tipe instans	Versi EMR Amazon yang didukung minimum (5.x, 6.x, 7.x)
	c5n.xlarge	emr-5.20.0, emr-6.0.0, emr-7.0.0
	c5n.2xlarge	emr-5.20.0, emr-6.0.0, emr-7.0.0
	c5n.4xlarge	emr-5.20.0, emr-6.0.0, emr-7.0.0
	c5n.9xlarge	emr-5.20.0, emr-6.0.0, emr-7.0.0
	c5n.18xlarge	emr-5.20.0, emr-6.0.0, emr-7.0.0
	c6g.xlarge	emr-5.31.0, emr-6.1.0, emr-7.0.0
	c6g.2xlarge	emr-5.31.0, emr-6.1.0, emr-7.0.0
	c6g.4xlarge	emr-5.31.0, emr-6.1.0, emr-7.0.0
	c6g.8xlarge	emr-5.31.0, emr-6.1.0, emr-7.0.0
	c6g.12xlarge	emr-5.31.0, emr-6.1.0, emr-7.0.0
	c6g.16xlarge	emr-5.31.0, emr-6.1.0, emr-7.0.0
	c6gd.xlarge	emr-5.33.0, emr-6.3.0, emr-7.0.0

Kelas Instans	Tipe instans	Versi EMR Amazon yang didukung minimum (5.x, 6.x, 7.x)
	c6gd.2xlarge	emr-5.33.0, emr-6.3.0, emr-7.0.0
	c6gd.4xlarge	emr-5.33.0, emr-6.3.0, emr-7.0.0
	c6gd.8xlarge	emr-5.33.0, emr-6.3.0, emr-7.0.0
	c6gd.12xlarge	emr-5.33.0, emr-6.3.0, emr-7.0.0
	c6gd.16xlarge	emr-5.33.0, emr-6.3.0, emr-7.0.0
	c6gn.xlarge	emr-5.33.0, emr-6.3.0, emr-7.0.0
	c6gn.2xlarge	emr-5.33.0, emr-6.3.0, emr-7.0.0
	c6gn.4xlarge	emr-5.33.0, emr-6.3.0, emr-7.0.0
	c6gn.8xlarge	emr-5.33.0, emr-6.3.0, emr-7.0.0
	c6gn.12xlarge	emr-5.33.0, emr-6.3.0, emr-7.0.0
	c6gn.16xlarge	emr-5.33.0, emr-6.3.0, emr-7.0.0
	c6i.xlarge	emr-5.35.0, emr-6.6.0
	c6i.2xlarge	emr-5.35.0, emr-6.6.0

Kelas Instans	Tipe instans	Versi EMR Amazon yang didukung minimum (5.x, 6.x, 7.x)
	c6i.4xlarge	emr-5.35.0, emr-6.6.0
	c6i.8xlarge	emr-5.35.0, emr-6.6.0
	c6i.12xlarge	emr-5.35.0, emr-6.6.0
	c6i.16xlarge	emr-5.35.0, emr-6.6.0
	c6i.24xlarge	emr-5.35.0, emr-6.6.0
	c6i.32xlarge	emr-5.35.0, emr-6.6.0
	c6in.xlarge	emr-5.36.1, emr-6.10.0
	c6in.2xlarge	emr-5.36.1, emr-6.10.0
	c6in.4xlarge	emr-5.36.1, emr-6.10.0
	c6in.8xlarge	emr-5.36.1, emr-6.10.0
	c6in.12xlarge	emr-5.36.1, emr-6.10.0
	c6in.16xlarge	emr-5.36.1, emr-6.10.0
	c6in.24xlarge	emr-5.36.1, emr-6.10.0
	c6in.32xlarge	emr-5.36.1, emr-6.10.0
	c7g.xlarge	emr-5.36.1, emr-6.7.0, emr-7.0.0
	c7g.2xlarge	emr-5.36.1, emr-6.7.0, emr-7.0.0
	c7g.4xlarge	emr-5.36.1, emr-6.7.0, emr-7.0.0

Kelas Instans	Tipe instans	Versi EMR Amazon yang didukung minimum (5.x, 6.x, 7.x)
	c7g.8xlarge	emr-5.36.1, emr-6.7.0, emr-7.0.0
	c7g.12xlarge	emr-5.36.1, emr-6.7.0, emr-7.0.0
	c7g.16xlarge	emr-5.36.1, emr-6.7.0, emr-7.0.0
	c7i.xlarge	emr-5.36.1, emr-6.10.0
	c7i.2xlarge	emr-5.36.1, emr-6.10.0
	c7i.4xlarge	emr-5.36.1, emr-6.10.0
	c7i.8xlarge	emr-5.36.1, emr-6.10.0
	c7i.12xlarge	emr-5.36.1, emr-6.10.0
	c7i.16xlarge	emr-5.36.1, emr-6.10.0
	c7i.24xlarge	emr-5.36.1, emr-6.10.0
	c7i.48xlarge	emr-5.36.1, emr-6.10.0
Komputasi yang Dipercepat	g4dn.xlarge	emr-5.30.0, emr-6.0.0, emr-7.0.0
	g4dn.2xlarge	emr-5.30.0, emr-6.0.0, emr-7.0.0
	g4dn.4xlarge	emr-5.30.0, emr-6.0.0, emr-7.0.0
	g4dn.8xlarge	emr-5.30.0, emr-6.0.0, emr-7.0.0

Kelas Instans	Tipe instans	Versi EMR Amazon yang didukung minimum (5.x, 6.x, 7.x)
	g4dn.12xlarge	emr-5.30.0, emr-6.0.0, emr-7.0.0
	g4dn.16xlarge	emr-5.30.0, emr-6.0.0, emr-7.0.0
	g5.xlarge	emr-5.36.1, emr-6.9.0, emr-7.0.0
	g5.2xlarge	emr-5.36.1, emr-6.9.0, emr-7.0.0
	g5.4xlarge	emr-5.36.1, emr-6.9.0, emr-7.0.0
	g5.8xlarge	emr-5.36.1, emr-6.9.0, emr-7.0.0
	g5.12xlarge	emr-5.36.1, emr-6.9.0, emr-7.0.0
	g5.16xlarge	emr-5.36.1, emr-6.9.0, emr-7.0.0
	g5.24xlarge	emr-5.36.1, emr-6.9.0, emr-7.0.0
	g5.48xlarge	emr-5.36.1, emr-6.9.0, emr-7.0.0
Memori Dioptimalkan	r5.xlarge	emr-5.17.0, emr-6.0.0, emr-7.0.0
	r5.2xlarge	emr-5.17.0, emr-6.0.0, emr-7.0.0

Kelas Instans	Tipe instans	Versi EMR Amazon yang didukung minimum (5.x, 6.x, 7.x)
	r5.4xlarge	emr-5.17.0, emr-6.0.0, emr-7.0.0
	r5.8xlarge	emr-5.17.0, emr-6.0.0, emr-7.0.0
	r5.12xlarge	emr-5.17.0, emr-6.0.0, emr-7.0.0
	r5.16xlarge	emr-5.17.0, emr-6.0.0, emr-7.0.0
	r5.24xlarge	emr-5.17.0, emr-6.0.0, emr-7.0.0
	r5b.xlarge	emr-5.33.0, emr-6.3.0, emr-7.0.0
	r5b.2xlarge	emr-5.33.0, emr-6.3.0, emr-7.0.0
	r5b.4xlarge	emr-5.33.0, emr-6.3.0, emr-7.0.0
	r5b.8xlarge	emr-5.33.0, emr-6.3.0, emr-7.0.0
	r5b.12xlarge	emr-5.33.0, emr-6.3.0, emr-7.0.0
	r5b.16xlarge	emr-5.33.0, emr-6.3.0, emr-7.0.0
	r5b.24xlarge	emr-5.33.0, emr-6.3.0, emr-7.0.0

Kelas Instans	Tipe instans	Versi EMR Amazon yang didukung minimum (5.x, 6.x, 7.x)
	r5d.xlarge	emr-5.17.0, emr-6.0.0, emr-7.0.0
	r5d.2xlarge	emr-5.17.0, emr-6.0.0, emr-7.0.0
	r5d.4xlarge	emr-5.17.0, emr-6.0.0, emr-7.0.0
	r5d.8xlarge	emr-5.17.0, emr-6.0.0, emr-7.0.0
	r5d.12xlarge	emr-5.17.0, emr-6.0.0, emr-7.0.0
	r5d.16xlarge	emr-5.17.0, emr-6.0.0, emr-7.0.0
	r5d.24xlarge	emr-5.17.0, emr-6.0.0, emr-7.0.0
	r5dn.xlarge	emr-5.31.0, emr-6.1.0, emr-7.0.0
	r5dn.2xlarge	emr-5.31.0, emr-6.1.0, emr-7.0.0
	r5dn.4xlarge	emr-5.31.0, emr-6.1.0, emr-7.0.0
	r5dn.8xlarge	emr-5.31.0, emr-6.1.0, emr-7.0.0
	r5dn.12xlarge	emr-5.31.0, emr-6.1.0, emr-7.0.0

Kelas Instans	Tipe instans	Versi EMR Amazon yang didukung minimum (5.x, 6.x, 7.x)
	r5dn.16xlarge	emr-5.31.0, emr-6.1.0, emr-7.0.0
	r5dn.24xlarge	emr-5.31.0, emr-6.1.0, emr-7.0.0
	r5n.xlarge	emr-5.31.0, emr-6.1.0, emr-7.0.0
	r5n.2xlarge	emr-5.31.0, emr-6.1.0, emr-7.0.0
	r5n.4xlarge	emr-5.31.0, emr-6.1.0, emr-7.0.0
	r5n.8xlarge	emr-5.31.0, emr-6.1.0, emr-7.0.0
	r5n.12xlarge	emr-5.31.0, emr-6.1.0, emr-7.0.0
	r5n.16xlarge	emr-5.31.0, emr-6.1.0, emr-7.0.0
	r5n.24xlarge	emr-5.31.0, emr-6.1.0, emr-7.0.0
	r6g.xlarge	emr-5.31.0, emr-6.1.0, emr-7.0.0
	r6g.2xlarge	emr-5.31.0, emr-6.1.0, emr-7.0.0
	r6g.4xlarge	emr-5.31.0, emr-6.1.0, emr-7.0.0

Kelas Instans	Tipe instans	Versi EMR Amazon yang didukung minimum (5.x, 6.x, 7.x)
	r6g.8xlarge	emr-5.31.0, emr-6.1.0, emr-7.0.0
	r6g.12xlarge	emr-5.31.0, emr-6.1.0, emr-7.0.0
	r6g.16xlarge	emr-5.31.0, emr-6.1.0, emr-7.0.0
	r6gd.xlarge	emr-5.33.0, emr-6.3.0, emr-7.0.0
	r6gd.2xlarge	emr-5.33.0, emr-6.3.0, emr-7.0.0
	r6gd.4xlarge	emr-5.33.0, emr-6.3.0, emr-7.0.0
	r6gd.8xlarge	emr-5.33.0, emr-6.3.0, emr-7.0.0
	r6gd.12xlarge	emr-5.33.0, emr-6.3.0, emr-7.0.0
	r6gd.16xlarge	emr-5.33.0, emr-6.3.0, emr-7.0.0
	r6i.xlarge	emr-5.35.0, emr-6.6.0
	r6i.2xlarge	emr-5.35.0, emr-6.6.0
	r6i.4xlarge	emr-5.35.0, emr-6.6.0
	r6i.8xlarge	emr-5.35.0, emr-6.6.0
	r6i.12xlarge	emr-5.35.0, emr-6.6.0

Kelas Instans	Tipe instans	Versi EMR Amazon yang didukung minimum (5.x, 6.x, 7.x)
	r6i.16xlarge	emr-5.35.0, emr-6.6.0
	r6i.24xlarge	emr-5.35.0, emr-6.6.0
	r6i.32xlarge	emr-5.35.0, emr-6.6.0
	r7g.xlarge	emr-5.36.1, emr-6.10.0, emr-7.0.0
	r7g.2xlarge	emr-5.36.1, emr-6.10.0, emr-7.0.0
	r7g.4xlarge	emr-5.36.1, emr-6.10.0, emr-7.0.0
	r7g.8xlarge	emr-5.36.1, emr-6.10.0, emr-7.0.0
	r7g.12xlarge	emr-5.36.1, emr-6.10.0, emr-7.0.0
	r7g.16xlarge	emr-5.36.1, emr-6.10.0, emr-7.0.0
	r7i.xlarge	emr-5.36.1, emr-6.10.0
	r7i.2xlarge	emr-5.36.1, emr-6.10.0
	r7i.4xlarge	emr-5.36.1, emr-6.10.0
	r7i.8xlarge	emr-5.36.1, emr-6.10.0
	r7i.12xlarge	emr-5.36.1, emr-6.10.0
	r7i.16xlarge	emr-5.36.1, emr-6.10.0
	r7i.24xlarge	emr-5.36.1, emr-6.10.0

Kelas Instans	Tipe instans	Versi EMR Amazon yang didukung minimum (5.x, 6.x, 7.x)
	r7i.48xlarge	emr-5.36.1, emr-6.10.0
	x2idn.16xlarge	emr-5.36.1, emr-6.9.0
	x2idn.24xlarge	emr-5.36.1, emr-6.9.0
	x2idn.32xlarge	emr-5.36.1, emr-6.9.0
	x2iedn.xlarge	emr-5.36.1, emr-6.9.0
	x2iedn.2xlarge	emr-5.36.1, emr-6.9.0
	x2iedn.4xlarge	emr-5.36.1, emr-6.9.0
	x2iedn.8xlarge	emr-5.36.1, emr-6.9.0
	x2iedn.16xlarge	emr-5.36.1, emr-6.9.0
	x2iedn.24xlarge	emr-5.36.1, emr-6.9.0
	x2iedn.32xlarge	emr-5.36.1, emr-6.9.0
Penyimpanan Dioptimalkan	i3.xlarge	emr-5.17.0, emr-6.0.0, emr-7.0.0
	i3.2xlarge	emr-5.17.0, emr-6.0.0, emr-7.0.0
	i3.4xlarge	emr-5.17.0, emr-6.0.0, emr-7.0.0
	i3.8xlarge	emr-5.17.0, emr-6.0.0, emr-7.0.0
	i3.16xlarge	emr-5.17.0, emr-6.0.0, emr-7.0.0

Kelas Instans	Tipe instans	Versi EMR Amazon yang didukung minimum (5.x, 6.x, 7.x)
	i3en.xlarge	emr-5.25.0, emr-6.0.0, emr-7.0.0
	i3en.2xlarge	emr-5.25.0, emr-6.0.0, emr-7.0.0
	i3en.3xlarge	emr-5.25.0, emr-6.0.0, emr-7.0.0
	i3en.6xlarge	emr-5.25.0, emr-6.0.0, emr-7.0.0
	i3en.12xlarge	emr-5.25.0, emr-6.0.0, emr-7.0.0
	i3en.24xlarge	emr-5.25.0, emr-6.0.0, emr-7.0.0
	i4i.xlarge	emr-5.36.1, emr-6.8.0
	i4i.2xlarge	emr-5.36.1, emr-6.8.0
	i4i.4xlarge	emr-5.36.1, emr-6.8.0
	i4i.8xlarge	emr-5.36.1, emr-6.8.0
	i4i.12xlarge	emr-5.36.1, emr-6.10.0
	i4i.16xlarge	emr-5.36.1, emr-6.8.0
	i4i.24xlarge	emr-5.36.1, emr-6.10.0
	i4i.32xlarge	emr-5.36.1, emr-6.8.0

Timur Tengah (Bahrain) - saya-selatan-1

Kelas Instans	Tipe instans	Versi EMR Amazon yang didukung minimum (5.x, 6.x, 7.x)
Tujuan Umum	m5.xlarge	emr-5.24.0, emr-6.0.0, emr-7.0.0
	m5.2xlarge	emr-5.24.0, emr-6.0.0, emr-7.0.0
	m5.4xlarge	emr-5.24.0, emr-6.0.0, emr-7.0.0
	m5.8xlarge	emr-5.24.0, emr-6.0.0, emr-7.0.0
	m5.12xlarge	emr-5.24.0, emr-6.0.0, emr-7.0.0
	m5.16xlarge	emr-5.24.0, emr-6.0.0, emr-7.0.0
	m5.24xlarge	emr-5.24.0, emr-6.0.0, emr-7.0.0
	m5d.xlarge	emr-5.24.0, emr-6.0.0, emr-7.0.0
	m5d.2xlarge	emr-5.24.0, emr-6.0.0, emr-7.0.0
	m5d.4xlarge	emr-5.24.0, emr-6.0.0, emr-7.0.0
	m5d.8xlarge	emr-5.24.0, emr-6.0.0, emr-7.0.0

Kelas Instans	Tipe instans	Versi EMR Amazon yang didukung minimum (5.x, 6.x, 7.x)
	m5d.12xlarge	emr-5.24.0, emr-6.0.0, emr-7.0.0
	m5d.16xlarge	emr-5.24.0, emr-6.0.0, emr-7.0.0
	m5d.24xlarge	emr-5.24.0, emr-6.0.0, emr-7.0.0
	m6g.xlarge	emr-5.30.0, emr-6.1.0, emr-7.0.0
	m6g.2xlarge	emr-5.30.0, emr-6.1.0, emr-7.0.0
	m6g.4xlarge	emr-5.30.0, emr-6.1.0, emr-7.0.0
	m6g.8xlarge	emr-5.30.0, emr-6.1.0, emr-7.0.0
	m6g.12xlarge	emr-5.30.0, emr-6.1.0, emr-7.0.0
	m6g.16xlarge	emr-5.30.0, emr-6.1.0, emr-7.0.0
	m6i.xlarge	emr-5.35.0, emr-6.6.0
	m6i.2xlarge	emr-5.35.0, emr-6.6.0
	m6i.4xlarge	emr-5.35.0, emr-6.6.0
	m6i.8xlarge	emr-5.35.0, emr-6.6.0
	m6i.12xlarge	emr-5.35.0, emr-6.6.0

Kelas Instans	Tipe instans	Versi EMR Amazon yang didukung minimum (5.x, 6.x, 7.x)
	m6i.16xlarge	emr-5.35.0, emr-6.6.0
	m6i.24xlarge	emr-5.35.0, emr-6.6.0
	m6i.32xlarge	emr-5.35.0, emr-6.6.0
Komputasi Dioptimalkan	c5.xlarge	emr-5.24.0, emr-6.0.0, emr-7.0.0
	c5.2xlarge	emr-5.24.0, emr-6.0.0, emr-7.0.0
	c5.4xlarge	emr-5.24.0, emr-6.0.0, emr-7.0.0
	c5.9xlarge	emr-5.24.0, emr-6.0.0, emr-7.0.0
	c5.12xlarge	emr-5.24.0, emr-6.0.0, emr-7.0.0
	c5.18xlarge	emr-5.24.0, emr-6.0.0, emr-7.0.0
	c5.24xlarge	emr-5.24.0, emr-6.0.0, emr-7.0.0
	c5a.xlarge	emr-5.31.0, emr-6.1.0, emr-7.0.0
	c5a.2xlarge	emr-5.31.0, emr-6.1.0, emr-7.0.0
	c5a.4xlarge	emr-5.31.0, emr-6.1.0, emr-7.0.0

Kelas Instans	Tipe instans	Versi EMR Amazon yang didukung minimum (5.x, 6.x, 7.x)
	c5a.8xlarge	emr-5.31.0, emr-6.1.0, emr-7.0.0
	c5a.12xlarge	emr-5.31.0, emr-6.1.0, emr-7.0.0
	c5a.16xlarge	emr-5.31.0, emr-6.1.0, emr-7.0.0
	c5a.24xlarge	emr-5.31.0, emr-6.1.0, emr-7.0.0
	c5ad.xlarge	emr-5.33.0, emr-6.3.0, emr-7.0.0
	c5ad.2xlarge	emr-5.33.0, emr-6.3.0, emr-7.0.0
	c5ad.4xlarge	emr-5.33.0, emr-6.3.0, emr-7.0.0
	c5ad.8xlarge	emr-5.33.0, emr-6.3.0, emr-7.0.0
	c5ad.12xlarge	emr-5.33.0, emr-6.3.0, emr-7.0.0
	c5ad.16xlarge	emr-5.33.0, emr-6.3.0, emr-7.0.0
	c5ad.24xlarge	emr-5.33.0, emr-6.3.0, emr-7.0.0
	c5d.xlarge	emr-5.24.0, emr-6.0.0, emr-7.0.0

Kelas Instans	Tipe instans	Versi EMR Amazon yang didukung minimum (5.x, 6.x, 7.x)
	c5d.2xlarge	emr-5.24.0, emr-6.0.0, emr-7.0.0
	c5d.4xlarge	emr-5.24.0, emr-6.0.0, emr-7.0.0
	c5d.9xlarge	emr-5.24.0, emr-6.0.0, emr-7.0.0
	c5d.18xlarge	emr-5.24.0, emr-6.0.0, emr-7.0.0
	c5n.xlarge	emr-5.24.0, emr-6.0.0, emr-7.0.0
	c5n.2xlarge	emr-5.24.0, emr-6.0.0, emr-7.0.0
	c5n.4xlarge	emr-5.24.0, emr-6.0.0, emr-7.0.0
	c5n.9xlarge	emr-5.24.0, emr-6.0.0, emr-7.0.0
	c5n.18xlarge	emr-5.24.0, emr-6.0.0, emr-7.0.0
	c6g.xlarge	emr-5.31.0, emr-6.1.0, emr-7.0.0
	c6g.2xlarge	emr-5.31.0, emr-6.1.0, emr-7.0.0
	c6g.4xlarge	emr-5.31.0, emr-6.1.0, emr-7.0.0

Kelas Instans	Tipe instans	Versi EMR Amazon yang didukung minimum (5.x, 6.x, 7.x)
	c6g.8xlarge	emr-5.31.0, emr-6.1.0, emr-7.0.0
	c6g.12xlarge	emr-5.31.0, emr-6.1.0, emr-7.0.0
	c6g.16xlarge	emr-5.31.0, emr-6.1.0, emr-7.0.0
	c6gn.xlarge	emr-5.33.0, emr-6.3.0, emr-7.0.0
	c6gn.2xlarge	emr-5.33.0, emr-6.3.0, emr-7.0.0
	c6gn.4xlarge	emr-5.33.0, emr-6.3.0, emr-7.0.0
	c6gn.8xlarge	emr-5.33.0, emr-6.3.0, emr-7.0.0
	c6gn.12xlarge	emr-5.33.0, emr-6.3.0, emr-7.0.0
	c6gn.16xlarge	emr-5.33.0, emr-6.3.0, emr-7.0.0
	c6i.xlarge	emr-5.35.0, emr-6.6.0
	c6i.2xlarge	emr-5.35.0, emr-6.6.0
	c6i.4xlarge	emr-5.35.0, emr-6.6.0
	c6i.8xlarge	emr-5.35.0, emr-6.6.0
	c6i.12xlarge	emr-5.35.0, emr-6.6.0

Kelas Instans	Tipe instans	Versi EMR Amazon yang didukung minimum (5.x, 6.x, 7.x)
	c6i.16xlarge	emr-5.35.0, emr-6.6.0
	c6i.24xlarge	emr-5.35.0, emr-6.6.0
	c6i.32xlarge	emr-5.35.0, emr-6.6.0
	c6in.xlarge	emr-5.36.1, emr-6.10.0
	c6in.2xlarge	emr-5.36.1, emr-6.10.0
	c6in.4xlarge	emr-5.36.1, emr-6.10.0
	c6in.8xlarge	emr-5.36.1, emr-6.10.0
	c6in.12xlarge	emr-5.36.1, emr-6.10.0
	c6in.16xlarge	emr-5.36.1, emr-6.10.0
	c6in.24xlarge	emr-5.36.1, emr-6.10.0
	c6in.32xlarge	emr-5.36.1, emr-6.10.0
Komputasi yang Dipercepat	g4dn.xlarge	emr-5.30.0, emr-6.0.0, emr-7.0.0
	g4dn.2xlarge	emr-5.30.0, emr-6.0.0, emr-7.0.0
	g4dn.4xlarge	emr-5.30.0, emr-6.0.0, emr-7.0.0
	g4dn.8xlarge	emr-5.30.0, emr-6.0.0, emr-7.0.0
	g4dn.12xlarge	emr-5.30.0, emr-6.0.0, emr-7.0.0

Kelas Instans	Tipe instans	Versi EMR Amazon yang didukung minimum (5.x, 6.x, 7.x)
	g4dn.16xlarge	emr-5.30.0, emr-6.0.0, emr-7.0.0
Memori Dioptimalkan	r5.xlarge	emr-5.24.0, emr-6.0.0, emr-7.0.0
	r5.2xlarge	emr-5.24.0, emr-6.0.0, emr-7.0.0
	r5.4xlarge	emr-5.24.0, emr-6.0.0, emr-7.0.0
	r5.8xlarge	emr-5.24.0, emr-6.0.0, emr-7.0.0
	r5.12xlarge	emr-5.24.0, emr-6.0.0, emr-7.0.0
	r5.16xlarge	emr-5.24.0, emr-6.0.0, emr-7.0.0
	r5.24xlarge	emr-5.24.0, emr-6.0.0, emr-7.0.0
	r5d.xlarge	emr-5.24.0, emr-6.0.0, emr-7.0.0
	r5d.2xlarge	emr-5.24.0, emr-6.0.0, emr-7.0.0
	r5d.4xlarge	emr-5.24.0, emr-6.0.0, emr-7.0.0
	r5d.8xlarge	emr-5.24.0, emr-6.0.0, emr-7.0.0

Kelas Instans	Tipe instans	Versi EMR Amazon yang didukung minimum (5.x, 6.x, 7.x)
	r5d.12xlarge	emr-5.24.0, emr-6.0.0, emr-7.0.0
	r5d.16xlarge	emr-5.24.0, emr-6.0.0, emr-7.0.0
	r5d.24xlarge	emr-5.24.0, emr-6.0.0, emr-7.0.0
	r6g.xlarge	emr-5.31.0, emr-6.1.0, emr-7.0.0
	r6g.2xlarge	emr-5.31.0, emr-6.1.0, emr-7.0.0
	r6g.4xlarge	emr-5.31.0, emr-6.1.0, emr-7.0.0
	r6g.8xlarge	emr-5.31.0, emr-6.1.0, emr-7.0.0
	r6g.12xlarge	emr-5.31.0, emr-6.1.0, emr-7.0.0
	r6g.16xlarge	emr-5.31.0, emr-6.1.0, emr-7.0.0
	r6i.xlarge	emr-5.35.0, emr-6.6.0
	r6i.2xlarge	emr-5.35.0, emr-6.6.0
	r6i.4xlarge	emr-5.35.0, emr-6.6.0
	r6i.8xlarge	emr-5.35.0, emr-6.6.0
	r6i.12xlarge	emr-5.35.0, emr-6.6.0

Kelas Instans	Tipe instans	Versi EMR Amazon yang didukung minimum (5.x, 6.x, 7.x)
	r6i.16xlarge	emr-5.35.0, emr-6.6.0
	r6i.24xlarge	emr-5.35.0, emr-6.6.0
	r6i.32xlarge	emr-5.35.0, emr-6.6.0
Penyimpanan Dioptimalkan	i3.xlarge	emr-5.24.0, emr-6.0.0, emr-7.0.0
	i3.2xlarge	emr-5.24.0, emr-6.0.0, emr-7.0.0
	i3.4xlarge	emr-5.24.0, emr-6.0.0, emr-7.0.0
	i3.8xlarge	emr-5.24.0, emr-6.0.0, emr-7.0.0
	i3.16xlarge	emr-5.24.0, emr-6.0.0, emr-7.0.0
	i3en.xlarge	emr-5.25.0, emr-6.0.0, emr-7.0.0
	i3en.2xlarge	emr-5.25.0, emr-6.0.0, emr-7.0.0
	i3en.3xlarge	emr-5.25.0, emr-6.0.0, emr-7.0.0
	i3en.6xlarge	emr-5.25.0, emr-6.0.0, emr-7.0.0
	i3en.12xlarge	emr-5.25.0, emr-6.0.0, emr-7.0.0

Kelas Instans	Tipe instans	Versi EMR Amazon yang didukung minimum (5.x, 6.x, 7.x)
	i3en.24xlarge	emr-5.25.0, emr-6.0.0, emr-7.0.0
	i4i.xlarge	emr-5.36.1, emr-6.8.0
	i4i.2xlarge	emr-5.36.1, emr-6.8.0
	i4i.4xlarge	emr-5.36.1, emr-6.8.0
	i4i.8xlarge	emr-5.36.1, emr-6.8.0
	i4i.12xlarge	emr-5.36.1, emr-6.10.0
	i4i.16xlarge	emr-5.36.1, emr-6.8.0
	i4i.24xlarge	emr-5.36.1, emr-6.10.0
	i4i.32xlarge	emr-5.36.1, emr-6.8.0

Timur Tengah (UEA) - saya-sentral-1

Kelas Instans	Tipe instans	Versi EMR Amazon yang didukung minimum (5.x, 6.x, 7.x)
Tujuan Umum	m5.xlarge	emr-5.36.0, emr-6.7.0, emr-7.0.0
	m5.2xlarge	emr-5.36.0, emr-6.7.0, emr-7.0.0
	m5.4xlarge	emr-5.36.0, emr-6.7.0, emr-7.0.0

Kelas Instans	Tipe instans	Versi EMR Amazon yang didukung minimum (5.x, 6.x, 7.x)
	m5.8xlarge	emr-5.36.0, emr-6.7.0, emr-7.0.0
	m5.12xlarge	emr-5.36.0, emr-6.7.0, emr-7.0.0
	m5.16xlarge	emr-5.36.0, emr-6.7.0, emr-7.0.0
	m5.24xlarge	emr-5.36.0, emr-6.7.0, emr-7.0.0
	m5d.xlarge	emr-5.36.0, emr-6.7.0, emr-7.0.0
	m5d.2xlarge	emr-5.36.0, emr-6.7.0, emr-7.0.0
	m5d.4xlarge	emr-5.36.0, emr-6.7.0, emr-7.0.0
	m5d.8xlarge	emr-5.36.0, emr-6.7.0, emr-7.0.0
	m5d.12xlarge	emr-5.36.0, emr-6.7.0, emr-7.0.0
	m5d.16xlarge	emr-5.36.0, emr-6.7.0, emr-7.0.0
	m5d.24xlarge	emr-5.36.0, emr-6.7.0, emr-7.0.0
	m6g.xlarge	emr-5.36.0, emr-6.7.0, emr-7.0.0

Kelas Instans	Tipe instans	Versi EMR Amazon yang didukung minimum (5.x, 6.x, 7.x)
	m6g.2xlarge	emr-5.36.0, emr-6.7.0, emr-7.0.0
	m6g.4xlarge	emr-5.36.0, emr-6.7.0, emr-7.0.0
	m6g.8xlarge	emr-5.36.0, emr-6.7.0, emr-7.0.0
	m6g.12xlarge	emr-5.36.0, emr-6.7.0, emr-7.0.0
	m6g.16xlarge	emr-5.36.0, emr-6.7.0, emr-7.0.0
	m6gd.xlarge	emr-5.36.0, emr-6.7.0, emr-7.0.0
	m6gd.2xlarge	emr-5.36.0, emr-6.7.0, emr-7.0.0
	m6gd.4xlarge	emr-5.36.0, emr-6.7.0, emr-7.0.0
	m6gd.8xlarge	emr-5.36.0, emr-6.7.0, emr-7.0.0
	m6gd.12xlarge	emr-5.36.0, emr-6.7.0, emr-7.0.0
	m6gd.16xlarge	emr-5.36.0, emr-6.7.0, emr-7.0.0
	m6i.xlarge	emr-5.36.0, emr-6.7.0
	m6i.2xlarge	emr-5.36.0, emr-6.7.0

Kelas Instans	Tipe instans	Versi EMR Amazon yang didukung minimum (5.x, 6.x, 7.x)
	m6i.4xlarge	emr-5.36.0, emr-6.7.0
	m6i.8xlarge	emr-5.36.0, emr-6.7.0
	m6i.12xlarge	emr-5.36.0, emr-6.7.0
	m6i.16xlarge	emr-5.36.0, emr-6.7.0
	m6i.24xlarge	emr-5.36.0, emr-6.7.0
	m6i.32xlarge	emr-5.36.0, emr-6.7.0
Komputasi Dioptimalkan	c5.xlarge	emr-5.36.0, emr-6.7.0, emr-7.0.0
	c5.2xlarge	emr-5.36.0, emr-6.7.0, emr-7.0.0
	c5.4xlarge	emr-5.36.0, emr-6.7.0, emr-7.0.0
	c5.9xlarge	emr-5.36.0, emr-6.7.0, emr-7.0.0
	c5.12xlarge	emr-5.36.0, emr-6.7.0, emr-7.0.0
	c5.18xlarge	emr-5.36.0, emr-6.7.0, emr-7.0.0
	c5.24xlarge	emr-5.36.0, emr-6.7.0, emr-7.0.0
	c5d.xlarge	emr-5.36.0, emr-6.7.0, emr-7.0.0

Kelas Instans	Tipe instans	Versi EMR Amazon yang didukung minimum (5.x, 6.x, 7.x)
	c5d.2xlarge	emr-5.36.0, emr-6.7.0, emr-7.0.0
	c5d.4xlarge	emr-5.36.0, emr-6.7.0, emr-7.0.0
	c5d.9xlarge	emr-5.36.0, emr-6.7.0, emr-7.0.0
	c5d.12xlarge	emr-5.36.0, emr-6.7.0, emr-7.0.0
	c5d.18xlarge	emr-5.36.0, emr-6.7.0, emr-7.0.0
	c5d.24xlarge	emr-5.36.0, emr-6.7.0, emr-7.0.0
	c6g.xlarge	emr-5.36.0, emr-6.7.0, emr-7.0.0
	c6g.2xlarge	emr-5.36.0, emr-6.7.0, emr-7.0.0
	c6g.4xlarge	emr-5.36.0, emr-6.7.0, emr-7.0.0
	c6g.8xlarge	emr-5.36.0, emr-6.7.0, emr-7.0.0
	c6g.12xlarge	emr-5.36.0, emr-6.7.0, emr-7.0.0
	c6g.16xlarge	emr-5.36.0, emr-6.7.0, emr-7.0.0

Kelas Instans	Tipe instans	Versi EMR Amazon yang didukung minimum (5.x, 6.x, 7.x)
	c6in.xlarge	emr-5.36.1, emr-6.10.0
	c6in.2xlarge	emr-5.36.1, emr-6.10.0
	c6in.4xlarge	emr-5.36.1, emr-6.10.0
	c6in.8xlarge	emr-5.36.1, emr-6.10.0
	c6in.12xlarge	emr-5.36.1, emr-6.10.0
	c6in.16xlarge	emr-5.36.1, emr-6.10.0
	c6in.24xlarge	emr-5.36.1, emr-6.10.0
	c6in.32xlarge	emr-5.36.1, emr-6.10.0
Komputasi yang Dipercepat	g5.xlarge	emr-5.36.1, emr-6.9.0, emr-7.0.0
	g5.2xlarge	emr-5.36.1, emr-6.9.0, emr-7.0.0
	g5.4xlarge	emr-5.36.1, emr-6.9.0, emr-7.0.0
	g5.8xlarge	emr-5.36.1, emr-6.9.0, emr-7.0.0
	g5.12xlarge	emr-5.36.1, emr-6.9.0, emr-7.0.0
	g5.16xlarge	emr-5.36.1, emr-6.9.0, emr-7.0.0
	g5.24xlarge	emr-5.36.1, emr-6.9.0, emr-7.0.0

Kelas Instans	Tipe instans	Versi EMR Amazon yang didukung minimum (5.x, 6.x, 7.x)
	g5.48xlarge	emr-5.36.1, emr-6.9.0, emr-7.0.0
Memori Dioptimalkan	r5.xlarge	emr-5.36.0, emr-6.7.0, emr-7.0.0
	r5.2xlarge	emr-5.36.0, emr-6.7.0, emr-7.0.0
	r5.4xlarge	emr-5.36.0, emr-6.7.0, emr-7.0.0
	r5.8xlarge	emr-5.36.0, emr-6.7.0, emr-7.0.0
	r5.12xlarge	emr-5.36.0, emr-6.7.0, emr-7.0.0
	r5.16xlarge	emr-5.36.0, emr-6.7.0, emr-7.0.0
	r5.24xlarge	emr-5.36.0, emr-6.7.0, emr-7.0.0
	r5d.xlarge	emr-5.36.0, emr-6.7.0, emr-7.0.0
	r5d.2xlarge	emr-5.36.0, emr-6.7.0, emr-7.0.0
	r5d.4xlarge	emr-5.36.0, emr-6.7.0, emr-7.0.0
	r5d.8xlarge	emr-5.36.0, emr-6.7.0, emr-7.0.0

Kelas Instans	Tipe instans	Versi EMR Amazon yang didukung minimum (5.x, 6.x, 7.x)
	r5d.12xlarge	emr-5.36.0, emr-6.7.0, emr-7.0.0
	r5d.16xlarge	emr-5.36.0, emr-6.7.0, emr-7.0.0
	r5d.24xlarge	emr-5.36.0, emr-6.7.0, emr-7.0.0
	r6g.xlarge	emr-5.36.0, emr-6.7.0, emr-7.0.0
	r6g.2xlarge	emr-5.36.0, emr-6.7.0, emr-7.0.0
	r6g.4xlarge	emr-5.36.0, emr-6.7.0, emr-7.0.0
	r6g.8xlarge	emr-5.36.0, emr-6.7.0, emr-7.0.0
	r6g.12xlarge	emr-5.36.0, emr-6.7.0, emr-7.0.0
	r6g.16xlarge	emr-5.36.0, emr-6.7.0, emr-7.0.0
	r6i.xlarge	emr-5.36.0, emr-6.7.0
	r6i.2xlarge	emr-5.36.0, emr-6.7.0
	r6i.4xlarge	emr-5.36.0, emr-6.7.0
	r6i.8xlarge	emr-5.36.0, emr-6.7.0
	r6i.12xlarge	emr-5.36.0, emr-6.7.0

Kelas Instans	Tipe instans	Versi EMR Amazon yang didukung minimum (5.x, 6.x, 7.x)
	r6i.16xlarge	emr-5.36.0, emr-6.7.0
	r6i.24xlarge	emr-5.36.0, emr-6.7.0
	r6i.32xlarge	emr-5.36.0, emr-6.7.0
	x2idn.16xlarge	emr-5.36.1, emr-6.9.0
	x2idn.24xlarge	emr-5.36.1, emr-6.9.0
	x2idn.32xlarge	emr-5.36.1, emr-6.9.0
Penyimpanan Dioptimalkan	i3.xlarge	emr-5.36.0, emr-6.7.0, emr-7.0.0
	i3.2xlarge	emr-5.36.0, emr-6.7.0, emr-7.0.0
	i3.4xlarge	emr-5.36.0, emr-6.7.0, emr-7.0.0
	i3.8xlarge	emr-5.36.0, emr-6.7.0, emr-7.0.0
	i3.16xlarge	emr-5.36.0, emr-6.7.0, emr-7.0.0
	i3en.xlarge	emr-5.36.0, emr-6.7.0, emr-7.0.0
	i3en.2xlarge	emr-5.36.0, emr-6.7.0, emr-7.0.0
	i3en.3xlarge	emr-5.36.0, emr-6.7.0, emr-7.0.0

Kelas Instans	Tipe instans	Versi EMR Amazon yang didukung minimum (5.x, 6.x, 7.x)
	i3en.6xlarge	emr-5.36.0, emr-6.7.0, emr-7.0.0
	i3en.12xlarge	emr-5.36.0, emr-6.7.0, emr-7.0.0
	i3en.24xlarge	emr-5.36.0, emr-6.7.0, emr-7.0.0
	i4i.xlarge	emr-5.36.1, emr-6.8.0
	i4i.2xlarge	emr-5.36.1, emr-6.8.0
	i4i.4xlarge	emr-5.36.1, emr-6.8.0
	i4i.8xlarge	emr-5.36.1, emr-6.8.0
	i4i.12xlarge	emr-5.36.1, emr-6.10.0
	i4i.16xlarge	emr-5.36.1, emr-6.8.0
	i4i.24xlarge	emr-5.36.1, emr-6.10.0
	i4i.32xlarge	emr-5.36.1, emr-6.8.0

South America (São Paulo) - sa-east-1

Kelas Instans	Tipe instans	Versi EMR Amazon yang didukung minimum (5.x, 6.x, 7.x)
Tujuan Umum	m5.xlarge	emr-5.13.0, emr-6.0.0, emr-7.0.0

Kelas Instans	Tipe instans	Versi EMR Amazon yang didukung minimum (5.x, 6.x, 7.x)
	m5.2xlarge	emr-5.13.0, emr-6.0.0, emr-7.0.0
	m5.4xlarge	emr-5.13.0, emr-6.0.0, emr-7.0.0
	m5.8xlarge	emr-5.13.0, emr-6.0.0, emr-7.0.0
	m5.12xlarge	emr-5.13.0, emr-6.0.0, emr-7.0.0
	m5.16xlarge	emr-5.13.0, emr-6.0.0, emr-7.0.0
	m5.24xlarge	emr-5.13.0, emr-6.0.0, emr-7.0.0
	m5a.xlarge	emr-5.20.0, emr-6.0.0, emr-7.0.0
	m5a.2xlarge	emr-5.20.0, emr-6.0.0, emr-7.0.0
	m5a.4xlarge	emr-5.20.0, emr-6.0.0, emr-7.0.0
	m5a.8xlarge	emr-5.20.0, emr-6.0.0, emr-7.0.0
	m5a.12xlarge	emr-5.20.0, emr-6.0.0, emr-7.0.0
	m5a.16xlarge	emr-5.20.0, emr-6.0.0, emr-7.0.0

Kelas Instans	Tipe instans	Versi EMR Amazon yang didukung minimum (5.x, 6.x, 7.x)
	m5a.24xlarge	emr-5.20.0, emr-6.0.0, emr-7.0.0
	m5ad.xlarge	emr-5.20.0, emr-6.0.0, emr-7.0.0
	m5ad.2xlarge	emr-5.20.0, emr-6.0.0, emr-7.0.0
	m5ad.4xlarge	emr-5.20.0, emr-6.0.0, emr-7.0.0
	m5ad.8xlarge	emr-5.33.0, emr-6.3.0, emr-7.0.0
	m5ad.12xlarge	emr-5.20.0, emr-6.0.0, emr-7.0.0
	m5ad.16xlarge	emr-5.33.0, emr-6.3.0, emr-7.0.0
	m5ad.24xlarge	emr-5.20.0, emr-6.0.0, emr-7.0.0
	m5d.xlarge	emr-5.13.0, emr-6.0.0, emr-7.0.0
	m5d.2xlarge	emr-5.13.0, emr-6.0.0, emr-7.0.0
	m5d.4xlarge	emr-5.13.0, emr-6.0.0, emr-7.0.0
	m5d.8xlarge	emr-5.13.0, emr-6.0.0, emr-7.0.0

Kelas Instans	Tipe instans	Versi EMR Amazon yang didukung minimum (5.x, 6.x, 7.x)
	m5d.12xlarge	emr-5.13.0, emr-6.0.0, emr-7.0.0
	m5d.16xlarge	emr-5.13.0, emr-6.0.0, emr-7.0.0
	m5d.24xlarge	emr-5.13.0, emr-6.0.0, emr-7.0.0
	m5zn.xlarge	emr-5.33.0, emr-6.3.0, emr-7.0.0
	m5zn.2xlarge	emr-5.33.0, emr-6.3.0, emr-7.0.0
	m5zn.3xlarge	emr-5.33.0, emr-6.3.0, emr-7.0.0
	m5zn.6xlarge	emr-5.33.0, emr-6.3.0, emr-7.0.0
	m5zn.12xlarge	emr-5.33.0, emr-6.3.0, emr-7.0.0
	m6a.xlarge	emr-5.36.1, emr-6.8.0, emr-7.0.0
	m6a.2xlarge	emr-5.36.1, emr-6.8.0, emr-7.0.0
	m6a.4xlarge	emr-5.36.1, emr-6.8.0, emr-7.0.0
	m6a.8xlarge	emr-5.36.1, emr-6.8.0, emr-7.0.0

Kelas Instans	Tipe instans	Versi EMR Amazon yang didukung minimum (5.x, 6.x, 7.x)
	m6a.12xlarge	emr-5.36.1, emr-6.8.0, emr-7.0.0
	m6a.16xlarge	emr-5.36.1, emr-6.8.0, emr-7.0.0
	m6a.24xlarge	emr-5.36.1, emr-6.8.0, emr-7.0.0
	m6a.32xlarge	emr-5.36.1, emr-6.8.0, emr-7.0.0
	m6a.48xlarge	emr-5.36.1, emr-6.8.0, emr-7.0.0
	m6g.xlarge	emr-5.30.0, emr-6.1.0, emr-7.0.0
	m6g.2xlarge	emr-5.30.0, emr-6.1.0, emr-7.0.0
	m6g.4xlarge	emr-5.30.0, emr-6.1.0, emr-7.0.0
	m6g.8xlarge	emr-5.30.0, emr-6.1.0, emr-7.0.0
	m6g.12xlarge	emr-5.30.0, emr-6.1.0, emr-7.0.0
	m6g.16xlarge	emr-5.30.0, emr-6.1.0, emr-7.0.0
	m6gd.xlarge	emr-5.33.0, emr-6.3.0, emr-7.0.0

Kelas Instans	Tipe instans	Versi EMR Amazon yang didukung minimum (5.x, 6.x, 7.x)
	m6gd.2xlarge	emr-5.33.0, emr-6.3.0, emr-7.0.0
	m6gd.4xlarge	emr-5.33.0, emr-6.3.0, emr-7.0.0
	m6gd.8xlarge	emr-5.33.0, emr-6.3.0, emr-7.0.0
	m6gd.12xlarge	emr-5.33.0, emr-6.3.0, emr-7.0.0
	m6gd.16xlarge	emr-5.33.0, emr-6.3.0, emr-7.0.0
	m6i.xlarge	emr-5.35.0, emr-6.6.0
	m6i.2xlarge	emr-5.35.0, emr-6.6.0
	m6i.4xlarge	emr-5.35.0, emr-6.6.0
	m6i.8xlarge	emr-5.35.0, emr-6.6.0
	m6i.12xlarge	emr-5.35.0, emr-6.6.0
	m6i.16xlarge	emr-5.35.0, emr-6.6.0
	m6i.24xlarge	emr-5.35.0, emr-6.6.0
	m6i.32xlarge	emr-5.35.0, emr-6.6.0
Komputasi Dioptimalkan	c5.xlarge	emr-5.13.0, emr-6.0.0, emr-7.0.0
	c5.2xlarge	emr-5.13.0, emr-6.0.0, emr-7.0.0

Kelas Instans	Tipe instans	Versi EMR Amazon yang didukung minimum (5.x, 6.x, 7.x)
	c5.4xlarge	emr-5.13.0, emr-6.0.0, emr-7.0.0
	c5.9xlarge	emr-5.13.0, emr-6.0.0, emr-7.0.0
	c5.12xlarge	emr-5.13.0, emr-6.0.0, emr-7.0.0
	c5.18xlarge	emr-5.13.0, emr-6.0.0, emr-7.0.0
	c5.24xlarge	emr-5.13.0, emr-6.0.0, emr-7.0.0
	c5a.xlarge	emr-5.31.0, emr-6.1.0, emr-7.0.0
	c5a.2xlarge	emr-5.31.0, emr-6.1.0, emr-7.0.0
	c5a.4xlarge	emr-5.31.0, emr-6.1.0, emr-7.0.0
	c5a.8xlarge	emr-5.31.0, emr-6.1.0, emr-7.0.0
	c5a.12xlarge	emr-5.31.0, emr-6.1.0, emr-7.0.0
	c5a.16xlarge	emr-5.31.0, emr-6.1.0, emr-7.0.0
	c5a.24xlarge	emr-5.31.0, emr-6.1.0, emr-7.0.0

Kelas Instans	Tipe instans	Versi EMR Amazon yang didukung minimum (5.x, 6.x, 7.x)
	c5ad.xlarge	emr-5.33.0, emr-6.3.0, emr-7.0.0
	c5ad.2xlarge	emr-5.33.0, emr-6.3.0, emr-7.0.0
	c5ad.4xlarge	emr-5.33.0, emr-6.3.0, emr-7.0.0
	c5ad.8xlarge	emr-5.33.0, emr-6.3.0, emr-7.0.0
	c5ad.12xlarge	emr-5.33.0, emr-6.3.0, emr-7.0.0
	c5ad.16xlarge	emr-5.33.0, emr-6.3.0, emr-7.0.0
	c5ad.24xlarge	emr-5.33.0, emr-6.3.0, emr-7.0.0
	c5d.xlarge	emr-5.13.0, emr-6.0.0, emr-7.0.0
	c5d.2xlarge	emr-5.13.0, emr-6.0.0, emr-7.0.0
	c5d.4xlarge	emr-5.13.0, emr-6.0.0, emr-7.0.0
	c5d.9xlarge	emr-5.13.0, emr-6.0.0, emr-7.0.0
	c5d.12xlarge	emr-5.13.0, emr-6.0.0, emr-7.0.0

Kelas Instans	Tipe instans	Versi EMR Amazon yang didukung minimum (5.x, 6.x, 7.x)
	c5d.18xlarge	emr-5.13.0, emr-6.0.0, emr-7.0.0
	c5d.24xlarge	emr-5.13.0, emr-6.0.0, emr-7.0.0
	c5n.xlarge	emr-5.20.0, emr-6.0.0, emr-7.0.0
	c5n.2xlarge	emr-5.20.0, emr-6.0.0, emr-7.0.0
	c5n.4xlarge	emr-5.20.0, emr-6.0.0, emr-7.0.0
	c5n.9xlarge	emr-5.20.0, emr-6.0.0, emr-7.0.0
	c5n.18xlarge	emr-5.20.0, emr-6.0.0, emr-7.0.0
	c6a.xlarge	emr-5.36.1, emr-6.8.0, emr-7.0.0
	c6a.2xlarge	emr-5.36.1, emr-6.8.0, emr-7.0.0
	c6a.4xlarge	emr-5.36.1, emr-6.8.0, emr-7.0.0
	c6a.8xlarge	emr-5.36.1, emr-6.8.0, emr-7.0.0
	c6a.12xlarge	emr-5.36.1, emr-6.8.0, emr-7.0.0

Kelas Instans	Tipe instans	Versi EMR Amazon yang didukung minimum (5.x, 6.x, 7.x)
	c6a.16xlarge	emr-5.36.1, emr-6.8.0, emr-7.0.0
	c6a.24xlarge	emr-5.36.1, emr-6.8.0, emr-7.0.0
	c6a.32xlarge	emr-5.36.1, emr-6.8.0, emr-7.0.0
	c6a.48xlarge	emr-5.36.1, emr-6.8.0, emr-7.0.0
	c6g.xlarge	emr-5.31.0, emr-6.1.0, emr-7.0.0
	c6g.2xlarge	emr-5.31.0, emr-6.1.0, emr-7.0.0
	c6g.4xlarge	emr-5.31.0, emr-6.1.0, emr-7.0.0
	c6g.8xlarge	emr-5.31.0, emr-6.1.0, emr-7.0.0
	c6g.12xlarge	emr-5.31.0, emr-6.1.0, emr-7.0.0
	c6g.16xlarge	emr-5.31.0, emr-6.1.0, emr-7.0.0
	c6gd.xlarge	emr-5.33.0, emr-6.3.0, emr-7.0.0
	c6gd.2xlarge	emr-5.33.0, emr-6.3.0, emr-7.0.0

Kelas Instans	Tipe instans	Versi EMR Amazon yang didukung minimum (5.x, 6.x, 7.x)
	c6gd.4xlarge	emr-5.33.0, emr-6.3.0, emr-7.0.0
	c6gd.8xlarge	emr-5.33.0, emr-6.3.0, emr-7.0.0
	c6gd.12xlarge	emr-5.33.0, emr-6.3.0, emr-7.0.0
	c6gd.16xlarge	emr-5.33.0, emr-6.3.0, emr-7.0.0
	c6gn.xlarge	emr-5.33.0, emr-6.3.0, emr-7.0.0
	c6gn.2xlarge	emr-5.33.0, emr-6.3.0, emr-7.0.0
	c6gn.4xlarge	emr-5.33.0, emr-6.3.0, emr-7.0.0
	c6gn.8xlarge	emr-5.33.0, emr-6.3.0, emr-7.0.0
	c6gn.12xlarge	emr-5.33.0, emr-6.3.0, emr-7.0.0
	c6gn.16xlarge	emr-5.33.0, emr-6.3.0, emr-7.0.0
	c6i.xlarge	emr-5.35.0, emr-6.6.0
	c6i.2xlarge	emr-5.35.0, emr-6.6.0
	c6i.4xlarge	emr-5.35.0, emr-6.6.0

Kelas Instans	Tipe instans	Versi EMR Amazon yang didukung minimum (5.x, 6.x, 7.x)
	c6i.8xlarge	emr-5.35.0, emr-6.6.0
	c6i.12xlarge	emr-5.35.0, emr-6.6.0
	c6i.16xlarge	emr-5.35.0, emr-6.6.0
	c6i.24xlarge	emr-5.35.0, emr-6.6.0
	c6i.32xlarge	emr-5.35.0, emr-6.6.0
	c6in.xlarge	emr-5.36.1, emr-6.10.0
	c6in.2xlarge	emr-5.36.1, emr-6.10.0
	c6in.4xlarge	emr-5.36.1, emr-6.10.0
	c6in.8xlarge	emr-5.36.1, emr-6.10.0
	c6in.12xlarge	emr-5.36.1, emr-6.10.0
	c6in.16xlarge	emr-5.36.1, emr-6.10.0
	c6in.24xlarge	emr-5.36.1, emr-6.10.0
	c6in.32xlarge	emr-5.36.1, emr-6.10.0
Komputasi yang Dipercepat	g4dn.xlarge	emr-5.30.0, emr-6.0.0, emr-7.0.0
	g4dn.2xlarge	emr-5.30.0, emr-6.0.0, emr-7.0.0
	g4dn.4xlarge	emr-5.30.0, emr-6.0.0, emr-7.0.0
	g4dn.8xlarge	emr-5.30.0, emr-6.0.0, emr-7.0.0

Kelas Instans	Tipe instans	Versi EMR Amazon yang didukung minimum (5.x, 6.x, 7.x)
	g4dn.12xlarge	emr-5.30.0, emr-6.0.0, emr-7.0.0
	g4dn.16xlarge	emr-5.30.0, emr-6.0.0, emr-7.0.0
	g5.xlarge	emr-5.36.1, emr-6.9.0, emr-7.0.0
	g5.2xlarge	emr-5.36.1, emr-6.9.0, emr-7.0.0
	g5.4xlarge	emr-5.36.1, emr-6.9.0, emr-7.0.0
	g5.8xlarge	emr-5.36.1, emr-6.9.0, emr-7.0.0
	g5.12xlarge	emr-5.36.1, emr-6.9.0, emr-7.0.0
	g5.16xlarge	emr-5.36.1, emr-6.9.0, emr-7.0.0
	g5.24xlarge	emr-5.36.1, emr-6.9.0, emr-7.0.0
	g5.48xlarge	emr-5.36.1, emr-6.9.0, emr-7.0.0
Memori Dioptimalkan	r5.xlarge	emr-5.13.0, emr-6.0.0, emr-7.0.0
	r5.2xlarge	emr-5.13.0, emr-6.0.0, emr-7.0.0

Kelas Instans	Tipe instans	Versi EMR Amazon yang didukung minimum (5.x, 6.x, 7.x)
	r5.4xlarge	emr-5.13.0, emr-6.0.0, emr-7.0.0
	r5.8xlarge	emr-5.13.0, emr-6.0.0, emr-7.0.0
	r5.12xlarge	emr-5.13.0, emr-6.0.0, emr-7.0.0
	r5.16xlarge	emr-5.13.0, emr-6.0.0, emr-7.0.0
	r5.24xlarge	emr-5.13.0, emr-6.0.0, emr-7.0.0
	r5a.xlarge	emr-5.20.0, emr-6.0.0, emr-7.0.0
	r5a.2xlarge	emr-5.20.0, emr-6.0.0, emr-7.0.0
	r5a.4xlarge	emr-5.20.0, emr-6.0.0, emr-7.0.0
	r5a.8xlarge	emr-5.20.0, emr-6.0.0, emr-7.0.0
	r5a.12xlarge	emr-5.20.0, emr-6.0.0, emr-7.0.0
	r5a.16xlarge	emr-5.20.0, emr-6.0.0, emr-7.0.0
	r5a.24xlarge	emr-5.20.0, emr-6.0.0, emr-7.0.0

Kelas Instans	Tipe instans	Versi EMR Amazon yang didukung minimum (5.x, 6.x, 7.x)
	r5ad.xlarge	emr-5.20.0, emr-6.0.0, emr-7.0.0
	r5ad.2xlarge	emr-5.20.0, emr-6.0.0, emr-7.0.0
	r5ad.4xlarge	emr-5.20.0, emr-6.0.0, emr-7.0.0
	r5ad.8xlarge	emr-5.33.0, emr-6.3.0, emr-7.0.0
	r5ad.12xlarge	emr-5.20.0, emr-6.0.0, emr-7.0.0
	r5ad.16xlarge	emr-5.33.0, emr-6.3.0, emr-7.0.0
	r5ad.24xlarge	emr-5.20.0, emr-6.0.0, emr-7.0.0
	r5b.xlarge	emr-5.33.0, emr-6.3.0, emr-7.0.0
	r5b.2xlarge	emr-5.33.0, emr-6.3.0, emr-7.0.0
	r5b.4xlarge	emr-5.33.0, emr-6.3.0, emr-7.0.0
	r5b.8xlarge	emr-5.33.0, emr-6.3.0, emr-7.0.0
	r5b.12xlarge	emr-5.33.0, emr-6.3.0, emr-7.0.0

Kelas Instans	Tipe instans	Versi EMR Amazon yang didukung minimum (5.x, 6.x, 7.x)
	r5b.16xlarge	emr-5.33.0, emr-6.3.0, emr-7.0.0
	r5b.24xlarge	emr-5.33.0, emr-6.3.0, emr-7.0.0
	r5d.xlarge	emr-5.13.0, emr-6.0.0, emr-7.0.0
	r5d.2xlarge	emr-5.13.0, emr-6.0.0, emr-7.0.0
	r5d.4xlarge	emr-5.13.0, emr-6.0.0, emr-7.0.0
	r5d.8xlarge	emr-5.13.0, emr-6.0.0, emr-7.0.0
	r5d.12xlarge	emr-5.13.0, emr-6.0.0, emr-7.0.0
	r5d.16xlarge	emr-5.13.0, emr-6.0.0, emr-7.0.0
	r5d.24xlarge	emr-5.13.0, emr-6.0.0, emr-7.0.0
	r5n.xlarge	emr-5.31.0, emr-6.1.0, emr-7.0.0
	r5n.2xlarge	emr-5.31.0, emr-6.1.0, emr-7.0.0
	r5n.4xlarge	emr-5.31.0, emr-6.1.0, emr-7.0.0

Kelas Instans	Tipe instans	Versi EMR Amazon yang didukung minimum (5.x, 6.x, 7.x)
	r5n.8xlarge	emr-5.31.0, emr-6.1.0, emr-7.0.0
	r5n.12xlarge	emr-5.31.0, emr-6.1.0, emr-7.0.0
	r5n.16xlarge	emr-5.31.0, emr-6.1.0, emr-7.0.0
	r5n.24xlarge	emr-5.31.0, emr-6.1.0, emr-7.0.0
	r6g.xlarge	emr-5.31.0, emr-6.1.0, emr-7.0.0
	r6g.2xlarge	emr-5.31.0, emr-6.1.0, emr-7.0.0
	r6g.4xlarge	emr-5.31.0, emr-6.1.0, emr-7.0.0
	r6g.8xlarge	emr-5.31.0, emr-6.1.0, emr-7.0.0
	r6g.12xlarge	emr-5.31.0, emr-6.1.0, emr-7.0.0
	r6g.16xlarge	emr-5.31.0, emr-6.1.0, emr-7.0.0
	r6gd.xlarge	emr-5.33.0, emr-6.3.0, emr-7.0.0
	r6gd.2xlarge	emr-5.33.0, emr-6.3.0, emr-7.0.0

Kelas Instans	Tipe instans	Versi EMR Amazon yang didukung minimum (5.x, 6.x, 7.x)
	r6gd.4xlarge	emr-5.33.0, emr-6.3.0, emr-7.0.0
	r6gd.8xlarge	emr-5.33.0, emr-6.3.0, emr-7.0.0
	r6gd.12xlarge	emr-5.33.0, emr-6.3.0, emr-7.0.0
	r6gd.16xlarge	emr-5.33.0, emr-6.3.0, emr-7.0.0
	r6i.xlarge	emr-5.35.0, emr-6.6.0
	r6i.2xlarge	emr-5.35.0, emr-6.6.0
	r6i.4xlarge	emr-5.35.0, emr-6.6.0
	r6i.8xlarge	emr-5.35.0, emr-6.6.0
	r6i.12xlarge	emr-5.35.0, emr-6.6.0
	r6i.16xlarge	emr-5.35.0, emr-6.6.0
	r6i.24xlarge	emr-5.35.0, emr-6.6.0
	r6i.32xlarge	emr-5.35.0, emr-6.6.0
	x1.16xlarge	emr-5.36.1, emr-6.9.0, emr-7.0.0
	x1.32xlarge	emr-5.36.1, emr-6.9.0, emr-7.0.0
	x1e.xlarge	emr-5.36.1, emr-6.10.0, emr-7.0.0

Kelas Instans	Tipe instans	Versi EMR Amazon yang didukung minimum (5.x, 6.x, 7.x)
	x1e.2xlarge	emr-5.36.1, emr-6.10.0, emr-7.0.0
	x1e.4xlarge	emr-5.36.1, emr-6.10.0, emr-7.0.0
	x1e.8xlarge	emr-5.36.1, emr-6.10.0, emr-7.0.0
	x1e.16xlarge	emr-5.36.1, emr-6.10.0, emr-7.0.0
	x1e.32xlarge	emr-5.36.1, emr-6.10.0, emr-7.0.0
	x2idn.16xlarge	emr-5.36.1, emr-6.9.0
	x2idn.24xlarge	emr-5.36.1, emr-6.9.0
	x2idn.32xlarge	emr-5.36.1, emr-6.9.0
	x2iedn.xlarge	emr-5.36.1, emr-6.9.0
	x2iedn.2xlarge	emr-5.36.1, emr-6.9.0
	x2iedn.4xlarge	emr-5.36.1, emr-6.9.0
	x2iedn.8xlarge	emr-5.36.1, emr-6.9.0
	x2iedn.16xlarge	emr-5.36.1, emr-6.9.0
	x2iedn.24xlarge	emr-5.36.1, emr-6.9.0
	x2iedn.32xlarge	emr-5.36.1, emr-6.9.0
Penyimpanan Dioptimalkan	i3.xlarge	emr-5.9.0, emr-6.0.0, emr-7.0.0

Kelas Instans	Tipe instans	Versi EMR Amazon yang didukung minimum (5.x, 6.x, 7.x)
	i3.2xlarge	emr-5.9.0, emr-6.0.0, emr-7.0.0
	i3.4xlarge	emr-5.9.0, emr-6.0.0, emr-7.0.0
	i3.8xlarge	emr-5.9.0, emr-6.0.0, emr-7.0.0
	i3.16xlarge	emr-5.9.0, emr-6.0.0, emr-7.0.0
	i3en.xlarge	emr-5.25.0, emr-6.0.0, emr-7.0.0
	i3en.2xlarge	emr-5.25.0, emr-6.0.0, emr-7.0.0
	i3en.3xlarge	emr-5.25.0, emr-6.0.0, emr-7.0.0
	i3en.6xlarge	emr-5.25.0, emr-6.0.0, emr-7.0.0
	i3en.12xlarge	emr-5.25.0, emr-6.0.0, emr-7.0.0
	i3en.24xlarge	emr-5.25.0, emr-6.0.0, emr-7.0.0
	i4i.xlarge	emr-5.36.1, emr-6.8.0
	i4i.2xlarge	emr-5.36.1, emr-6.8.0
	i4i.4xlarge	emr-5.36.1, emr-6.8.0

Kelas Instans	Tipe instans	Versi EMR Amazon yang didukung minimum (5.x, 6.x, 7.x)
	i4i.8xlarge	emr-5.36.1, emr-6.8.0
	i4i.12xlarge	emr-5.36.1, emr-6.10.0
	i4i.16xlarge	emr-5.36.1, emr-6.8.0
	i4i.24xlarge	emr-5.36.1, emr-6.10.0
	i4i.32xlarge	emr-5.36.1, emr-6.8.0

Instans generasi sebelumnya

Amazon EMR mendukung instans generasi sebelumnya untuk mendukung aplikasi yang dioptimalkan untuk instance ini dan belum ditingkatkan. Untuk informasi selengkapnya tentang jenis instans ini dan path pembaruan, lihat Instans Generasi sebelumnya.

Kelas Instans	Tipe instans
General Purpose	m1.small[1] | m1.medium[1] | m1.large[1] | m1.xlarge[1] | m3.xlarge[1] | m3.2xlarge[1] | m4.large | m4.xlarge | m4.2xlarge | m4.4xlarge | m4.10xlarge | m4.16xlarge
Compute Optimized	c1.medium[1] [2] | c1.xlarge[1] | c3.xlarge[1] | c3.2xlarge[1] | c3.4xlarge[1] | c3.8xlarge[1] | c4.large | c4.xlarge | c4.2xlarge | c4.4xlarge | c4.8xlarge
Memory Optimized	m2.xlarge[1] | m2.2xlarge[1] | m2.4xlarge[1] | r3.xlarge | r3.2xlarge | r3.4xlarge | r3.8xlarge | r4.xlarge | r4.2xlarge | r4.4xlarge | r4.8xlarge | r4.16xlarge
Storage Optimized	d2.xlarge | d2.2xlarge | d2.4xlarge | d2.8xlarge | i2.xlarge | i2.2xlarge | i2.4xlarge | i2.8xlarge

[1] Menggunakan virtualisasi PVM AMI dengan versi rilis Amazon EMR lebih awal dari 5.13.0. Untuk informasi selengkapnya, lihat Jenis Virtualisasi AMI Linux.

[2] Tidak didukung dalam versi rilis 5.15.0.

Opsi pembelian instans

Saat menyiapkan klaster, Anda memilih opsi pembelian untuk instans Amazon EC2. Anda dapat memilih Instans Sesuai Permintaan, Instans Spot, atau keduanya. Harga berbeda-beda berdasarkan jenis instans dan Wilayah. Harga Amazon EMR merupakan tambahan dari harga Amazon EC2 (harga untuk server yang mendasarinya) dan harga Amazon EBS (jika melampirkan volume Amazon EBS). Untuk harga saat ini, lihat Harga Amazon EMR.

Pilihan Anda untuk menggunakan grup instans atau armada instans di klaster menentukan bagaimana Anda dapat mengubah opsi pembelian instans saat klaster sedang berjalan. Jika memilih grup instans seragam, Anda hanya dapat menentukan opsi pembelian untuk grup instans saat membuatnya, dan jenis instans serta opsi pembelian berlaku untuk semua instans Amazon EC2 di setiap grup instans. Jika Anda memilih armada instans, Anda dapat mengubah opsi pembelian setelah Anda membuat armada instans, lalu Anda dapat menggabungkan opsi pembelian untuk memenuhi kapasitas target yang Anda tentukan. Untuk informasi selengkapnya tentang konfigurasi ini, lihat Membuat sebuah klaster dengan armada instan atau grup instans seragam.

Instans Sesuai Permintaan

Dengan Instans Sesuai Permintaan, Anda membayar kapasitas komputasi per detik. Secara opsional, Anda dapat meminta Instans Sesuai Permintaan ini menggunakan opsi pembelian Instans Cadangan atau Instans Khusus. Dengan Instans Cadangan, Anda melakukan pembayaran satu kali untuk instans guna mencadangkan kapasitas. Instans khusus diisolasi secara fisik di tingkat perangkat keras host dari instans yang menjadi milik akun AWS lainnya. Untuk informasi selengkapnya tentang opsi pembelian, lihat Opsi Pembelian Instans di Panduan Pengguna Amazon EC2 untuk Instans Linux.

Menggunakan Instans Cadangan

Untuk menggunakan Instans Cadangan di Amazon EMR, gunakan Amazon EC2 untuk membeli Instans Cadangan dan menentukan parameter reservasi, termasuk cakupan reservasi yang berlaku untuk Wilayah atau Availability Zone. Untuk informasi selengkapnya, lihat Instans Cadangan Amazon EC2 dan Membeli Instans Cadangan di Panduan Pengguna Amazon EC2 untuk Instans Linux. Setelah Anda membeli Instans Cadangan, jika semua kondisi berikut ini benar, Amazon EMR akan menggunakan Instans Cadangan saat klaster diluncurkan:

- Instans Sesuai Permintaan ditentukan dalam konfigurasi klaster yang cocok dengan spesifikasi Instans Cadangan.
- Klaster diluncurkan dalam cakupan reservasi instans (Availability Zone atau Wilayah).
- Kapasitas Instans Cadangan masih tersedia

Misalnya, Anda membeli satu `m5.xlarge` Instans Cadangan dengan reservasi instans yang dicakup di Wilayah AS-Timur. Anda kemudian meluncurkan cluster EMR Amazon di AS-Timur yang menggunakan dua instance. `m5.xlarge` Instans pertama ditagih dengan tarif Instans Cadangan dan yang lainnya ditagih dengan tarif Sesuai Permintaan. Kapasitas Instans Cadangan digunakan sebelum Instans Sesuai Permintaan dibuat.

Menggunakan Instans Khusus

Untuk menggunakan Instans Khusus, beli Instans Khusus menggunakan Amazon EC2 lalu buat VPC dengan atribut penghunian khusus. Dalam Amazon EMR, lalu tentukan sebuah klaster yang harus diluncurkan di VPC ini. Setiap Instans Sesuai Permintaan dalam klaster yang cocok dengan spesifikasi Instans Khusus menggunakan Instans Khusus yang tersedia saat klaster diluncurkan.

> **Note**
>
> Amazon EMR tidak mendukung pengaturan `dedicated` atribut pada instans individual.

Instans Spot

Instans Spot di Amazon EMR memberikan opsi bagi Anda untuk membeli kapasitas instans Amazon EC2 dengan biaya yang lebih rendah dibandingkan dengan pembelian Sesuai Permintaan. Kerugian menggunakan Instans Spot adalah bahwa instans dapat diakhiri jika kapasitas Spot menjadi tidak tersedia untuk jenis instans yang Anda jalankan. Untuk informasi lebih lanjut tentang kapan menggunakan Instans Spot yang mungkin sesuai untuk aplikasi Anda, lihat Kapan Anda harus menggunakan Instans Spot?.

Jika Amazon EC2 memiliki kapasitas yang tidak digunakan, ia menawarkan instans EC2 dengan biaya lebih rendah, yang mana disebut Harga spot. Harga ini berfluktuasi berdasarkan ketersediaan dan permintaan, dan ditetapkan sesuai Wilayah dan Availability Zone. Saat Anda memilih Instans Spot, tentukan harga Spot maksimum yang bersedia Anda bayar untuk setiap tipe instans EC2. Jika harga Spot di klaster Availability Zone di bawah harga Spot maksimum yang ditentukan untuk tipe

instans tersebut, maka instans akan diluncurkan. Saat instans berjalan, Anda ditagih dengan harga Spot saat ini, bukan harga Spot maksimum Anda.

> **Note**
>
> Instans Spot dengan durasi yang ditentukan (juga dikenal sebagai blok Spot) tidak lagi tersedia untuk pelanggan baru mulai 1 Juli 2021. Untuk pelanggan yang sebelumnya telah menggunakan fitur ini, kami akan terus mendukung Instans Spot dengan durasi yang ditentukan hingga 31 Desember 2022.

Untuk harga saat ini, lihat Harga Instans Spot Amazon EC2. Untuk informasi selengkapnya, lihat Instans Spot dalam Panduan Pengguna Amazon EC2 untuk Instans Linux. Saat Anda membuat dan mengonfigurasi klaster, tentukan opsi jaringan yang pada akhirnya menentukan Availability Zone tempat klaster Anda diluncurkan. Untuk informasi selengkapnya, lihat Mengkonfigurasi jaringan.

> **Tip**
>
> Anda dapat melihat harga Spot waktu nyata di konsol saat Anda mengarahkan kursor ke tooltip informasi di sebelah opsi pembelian Spot sewaktu Anda membuat klaster menggunakan Opsi Lanjutan. Harga untuk setiap Availability Zone di Wilayah yang dipilih akan ditampilkan. Harga terendah ada di barisan berwarna hijau. Karena harga Spot yang berfluktuasi di antara Availability Zone, memilih Availability Zone dengan harga awal terendah mungkin tidak menghasilkan harga terendah selama masa pakai klaster. Untuk hasil yang optimal, pelajari riwayat harga Availability Zone sebelum memilih. Untuk informasi selengkapnya, lihat Riwayat Harga Instans Spot dalam Panduan Pengguna Amazon EC2 untuk Instans Linux.

Opsi Instans Spot bergantung pada apakah Anda menggunakan grup instans seragam atau armada instans dalam konfigurasi klaster Anda.

Spot Instance dalam grup instans seragam

Saat Anda menggunakan Instans Spot dalam grup instans seragam, semua instans dalam grup instans harus menjadi Instans Spot. Anda menentukan subnet tunggal atau Availability Zone untuk klaster. Untuk setiap grup instans, tenentukan satu Instans Spot dan harga Spot maksimum. Instans Spot dari jenis tersebut diluncurkan jika harga Spot di Wilayah dan Availability Zone klaster berada di bawah harga Spot maksimum. Instans berakhir jika harga Spot berada di atas harga Spot maksimum

Anda. Anda menetapkan harga Spot maksimum hanya saat Anda mengonfigurasi grup instans. Hal ini tidak dapat diubah nanti. Untuk informasi selengkapnya, lihat Membuat sebuah klaster dengan armada instan atau grup instans seragam.

Instans Spot di armada instance

Saat Anda menggunakan konfigurasi armada instans, opsi tambahan akan memberi kontrol lebih besar bagi Anda atas bagaimana Instans Spot diluncurkan dan diakhiri. Pada dasarnya, armada instans menggunakan metode yang berbeda dari grup instans seragam untuk meluncurkan instans. Cara kerjanya adalah Anda menetapkan kapasitas target untuk Instans Spot (dan Instans Sesuai Permintaan) dan hingga lima tipe instans. Anda juga dapat menentukan kapasitas tertimbang untuk setiap tipe instans atau menggunakan vCPU (Vcore YARN) dari tipe instans sebagai kapasitas tertimbang. Kapasitas tertimbang ini diperhitungkan dalam kapasitas target Anda saat instans dari tipe tersebut disediakan. Amazon EMR menyediakan instans dengan kedua opsi pembelian hingga kapasitas target untuk setiap target yang terpenuhi. Selain itu, Anda dapat menentukan serangkaian Availability Zone untuk Amazon EMR untuk dipilih saat meluncurkan instans. Anda juga menyediakan opsi Spot tambahan untuk setiap armada, termasuk batas waktu penyediaan. Untuk informasi selengkapnya, lihat Mengkonfigurasi armada instans.

Penyimpanan instans

Gambaran Umum

Penyimpanan instans dan penyimpanan volume Amazon EBS digunakan untuk data HDFS dan untuk buffer, cache, data awal, dan konten sementara lainnya yang mungkin “tumpah” oleh beberapa aplikasi ke sistem file lokal.

Amazon EBS bekerja secara berbeda dalam Amazon EMR dibandingkan dengan instans Amazon EC2 biasa. Volume Amazon EBS yang dilampirkan ke kluster EMR Amazon bersifat sementara: volume dihapus saat klaster dan penghentian instans (misalnya, saat mengecilkan grup instans), jadi Anda seharusnya tidak mengharapkan data tetap ada. Meskipun datanya fana, ada kemungkinan bahwa data dalam HDFS dapat direplikasi tergantung pada jumlah dan spesialisasi node di cluster. Saat Anda menambahkan volume penyimpanan Amazon EBS, volume ini dipasang sebagai volume tambahan. Mereka bukan bagian dari volume asal. YARN dikonfigurasi untuk menggunakan semua volume tambahan, tetapi Anda bertanggung jawab untuk mengalokasikan volume tambahan sebagai penyimpanan lokal (untuk file log lokal, misalnya).

Pertimbangan-pertimbangan

Ingatlah pertimbangan tambahan ini saat Anda menggunakan Amazon EBS dengan kluster EMR:

- Anda tidak dapat memotret volume Amazon EBS dan kemudian mengembalikannya dalam Amazon EMR. Untuk membuat konfigurasi kustom yang dapat digunakan kembali, gunakan AMI kustom (tersedia di Amazon EMR versi 5.7.0 dan yang lebih baru). Untuk informasi selengkapnya, lihat Menggunakan AMI kustom.
- Volume perangkat root Amazon EBS terenkripsi hanya didukung saat menggunakan AMI khusus. Untuk informasi selengkapnya, lihat Membuat AMI khusus dengan volume perangkat asal Amazon EBS terenkripsi.
- Jika Anda menerapkan tag menggunakan API Amazon EMR, operasi tersebut diterapkan ke volume EBS.
- Ada batas 25 volume per instans.
- Volume Amazon EBS pada node inti tidak boleh kurang dari 5 GB.

Penyimpanan Amazon EBS default untuk instans

Untuk instans EC2 yang memiliki penyimpanan khusus EBS, Amazon EMR mengalokasikan volume penyimpanan Amazon EBS gp2 atau gp3 ke instans. Saat Anda membuat klaster dengan Amazon EMR merilis 5.22.0 dan yang lebih tinggi, jumlah default penyimpanan Amazon EBS meningkat relatif terhadap ukuran instans.

Kami membagi penyimpanan yang meningkat di beberapa volume. Ini memberikan peningkatan kinerja IOPS dan, pada gilirannya, peningkatan kinerja untuk beberapa beban kerja standar. Jika Anda ingin menggunakan konfigurasi penyimpanan instans Amazon EBS yang berbeda, Anda dapat menentukan ini saat membuat klaster EMR atau menambahkan node ke cluster yang ada. Anda dapat menggunakan volume Amazon EBS gp2 atau gp3 sebagai volume root, dan menambahkan volume gp2 atau gp3 sebagai volume tambahan. Untuk informasi selengkapnya, lihat Menentukan volume penyimpanan EBS tambahan.

Tabel berikut mengidentifikasi jumlah default volume penyimpanan Amazon EBS gp2, ukuran, dan ukuran total per jenis instans. Untuk informasi tentang volume gp2 dibandingkan dengan gp3, lihat. Membandingkan jenis volume Amazon EBS gp2 dan gp3

Volume dan ukuran penyimpanan Amazon EBS gp2 default berdasarkan jenis instans untuk Amazon EMR 5.22.0 dan yang lebih tinggi

Ukuran instans	Jumlah volume	Ukuran volume (GiB)	Ukuran total (GiB)
*.large	1	32	32

Ukuran instans	Jumlah volume	Ukuran volume (GiB)	Ukuran total (GiB)
*.xlarge	2	32	64
*.2xlarge	4	32	128
*.4xlarge	4	64	256
*.8xlarge	4	128	512
*.9xlarge	4	144	576
*.10xlarge	4	160	640
12xlarge	4	192	768
*.16xlarge	4	256	1024
*.18xlarge	4	288	1152
*.24xlarge	4	384	1536

Volume root Amazon EBS default untuk instance

Dengan Amazon EMR rilis 6.15 dan lebih tinggi, Amazon EMR secara otomatis memasang Amazon EBS General Purpose SSD (gp3) sebagai perangkat root untuk AMI untuk meningkatkan kinerja. Dengan rilis sebelumnya, Amazon EMR melampirkan EBS General Purpose SSD (gp2) sebagai perangkat root.

	6.15 dan lebih tinggi	6.14 dan lebih rendah
Jenis volume root default		
Ukuran default		
IOPS standar		
Throughput default		

Untuk informasi tentang cara menyesuaikan volume perangkat root Amazon EBS, lihatMenentukan volume penyimpanan EBS tambahan.

Menentukan volume penyimpanan EBS tambahan

Saat Anda mengonfigurasi tipe instans di Amazon EMR, Anda dapat menentukan volume EBS tambahan untuk menambah kapasitas di luar penyimpanan instans (jika ada) dan volume EBS default. Amazon EBS menyediakan jenis volume berikut: General Purpose (SSD), Provisioned IOPS (SSD), Throughput Optimized (HDD), Cold (HDD), dan Magnetic. Mereka berbeda dalam karakteristik kinerja dan harga, sehingga Anda dapat menyesuaikan penyimpanan Anda dengan kebutuhan analitik dan bisnis aplikasi Anda. Misalnya, beberapa aplikasi mungkin perlu tumpah ke disk sementara yang lain dapat bekerja dengan aman di memori atau dengan Amazon S3.

Anda hanya dapat melampirkan volume Amazon EBS ke instans pada waktu startup cluster dan saat Anda menambahkan grup instance node tugas tambahan. Jika instance di kluster EMR Amazon gagal, maka instance dan volume Amazon EBS terlampir diganti dengan volume baru. Akibatnya, jika Anda melepaskan volume Amazon EBS secara manual, Amazon EMR menganggapnya sebagai kegagalan dan menggantikan penyimpanan instans (jika ada) dan penyimpanan volume.

Amazon EMR tidak memungkinkan Anda mengubah jenis volume dari gp2 ke gp3 untuk cluster EMR yang ada. Untuk menggunakan gp3 untuk beban kerja Anda, luncurkan klaster EMR baru. Selain itu, kami tidak menyarankan Anda memperbarui throughput dan IOPS pada klaster yang sedang digunakan atau yang sedang disediakan, karena Amazon EMR menggunakan throughput dan nilai IOPS yang Anda tentukan pada waktu peluncuran klaster untuk instance baru apa pun yang ditambahkan selama peningkatan skala cluster. Untuk informasi selengkapnya, silakan lihat Membandingkan jenis volume Amazon EBS gp2 dan gp3 dan Memilih IOPS dan throughput saat bermigrasi ke gp3.

> **Important**
>
> Untuk menggunakan volume gp3 dengan cluster EMR Anda, Anda harus meluncurkan cluster baru.

Membandingkan jenis volume Amazon EBS gp2 dan gp3

Berikut adalah perbandingan biaya antara volume gp2 dan gp3 di Wilayah AS Timur (Virginia N.). Untuk informasi terbaru, lihat halaman produk Volume Tujuan Umum Amazon EBS dan Halaman Harga Amazon EBS.

Tipe volume	gp3	gp2
Ukuran volume	1 GiB - 16 TiB	1 GiB - 16 TiB
IOPS Default/Baseline	3000	3 IOPS/GIB (minimal 100 IOPS) hingga maksimum 16.000 IOPS. Volume yang lebih kecil dari 1 TiB juga dapat meledak hingga 3.000 IOPS.
IOP/Volume Maks	16.000	16.000
Throughput default/Baseline	125 MiB/s	Batas throughput adalah antara 128 MiB/s dan 250 MiB/s, tergantung pada ukuran volume.
Throughput/volume maks	1.000 MiB/dtk	250 MiB/dtk
Harga	$0,08/GIB-bulan 3.000 IOPS gratis dan $0,005/bulan IOPS yang disediakan lebih dari 3.000; 125 MIB/s gratis dan $0,04/MIB/S bulan yang disediakan lebih dari 125 MiB/s	$0.10/GIB-bulan

Memilih IOPS dan throughput saat bermigrasi ke gp3

Saat menyediakan volume gp2, Anda harus mengetahui ukuran volume untuk mendapatkan IOPS dan throughput proporsional. Dengan gp3, Anda tidak perlu menyediakan volume yang lebih besar untuk mendapatkan kinerja yang lebih tinggi. Anda dapat memilih ukuran dan kinerja yang Anda inginkan sesuai dengan kebutuhan aplikasi. Memilih ukuran yang tepat dan parameter kinerja yang tepat (IOPS, throughput) dapat memberi Anda pengurangan biaya maksimum, tanpa memengaruhi kinerja.

Berikut adalah tabel untuk membantu Anda memilih opsi konfigurasi gp3:

Ukuran volume	IOPS	Throughput
1—170 GiB	3000	125 MiB/s
170—334 GiB	3000	125 MiB/s jika jenis instans EC2 yang dipilih mendukung 125MiB/s atau kurang, gunakan lebih tinggi sesuai penggunaan, Maks 250 MiB/*.
334—1000 GiB	3000	125 MiB/s jika jenis instans EC2 yang dipilih mendukung 125MiB/s atau kurang, Gunakan lebih tinggi sesuai penggunaan, Maks 250 MiB/*.
1000+ GiB	Cocokkan gp2 IOPS (Ukuran dalam GiB x 3) atau IOPS Maks yang digerakkan oleh volume gp2 saat ini	125 MiB/s jika jenis instans EC2 yang dipilih mendukung 125MiB/s atau kurang, Gunakan lebih tinggi sesuai penggunaan, Maks 250 MiB/*.

* Gp3 memiliki kemampuan untuk menyediakan throughput hingga 1000 MiB/s. Karena gp2 menyediakan throughput maksimum 250MiB/s, Anda mungkin tidak perlu melampaui batas ini saat Anda menggunakan gp3.

Mengkonfigurasi jaringan

Sebagian besar klaster diluncurkan ke jaringan virtual menggunakan Amazon Virtual Private Cloud (Amazon VPC). VPC adalah jaringan virtual terisolasi di dalam AWS yang secara logis terisolasi di dalam AWS akun Anda. Anda dapat mengonfigurasi aspek seperti rentang alamat IP privat, subnet, tabel rute, dan gateway jaringan. Untuk informasi selengkapnya, silakan lihat ACL Jaringan di Panduan Pengguna Amazon VPC.

VPC menawarkan kemampuan berikut:

- Memproses data sensitif

Meluncurkan klaster ke VPC mirip dengan meluncurkan klaster ke jaringan privat dengan alat tambahan, seperti tabel rute dan ACL jaringan, untuk menentukan siapa yang memiliki akses ke jaringan. Jika Anda sedang memproses data sensitif di klaster, Anda mungkin menginginkan kontrol akses tambahan yang meluncurkan klaster ke dalam VPC yang disediakan. Selanjutnya, Anda dapat memilih untuk meluncurkan sumber daya Anda ke subnet privat di mana tidak ada dari sumber daya tersebut yang memiliki konektivitas internet langsung.

- Mengakses sumber daya pada jaringan internal

  Jika sumber data Anda berada di jaringan privat, sumber data tersebut mungkin tidak praktis atau tidak diperlukan untuk mengunggah data tersebut ke AWS untuk diimpor ke Amazon EMR, baik karena jumlah data yang akan ditransfer atau karena sifat data yang sensitif. Sebaliknya, Anda dapat meluncurkan klaster ke VPC dan menghubungkan pusat data Anda ke VPC melalui koneksi VPN, yang memungkinkan klaster untuk mengakses sumber daya di jaringan internal Anda. Misalnya, jika Anda memiliki basis data Oracle di pusat data Anda, dengan meluncurkan klaster ke VPC yang terhubung ke jaringan tersebut melalui VPN memungkinkan klaster untuk mengakses basis data Oracle.

Subnet publik dan pribadi

Anda dapat meluncurkan kluster EMR Amazon di subnet VPC publik dan pribadi. Ini berarti Anda tidak memerlukan konektivitas internet untuk menjalankan kluster EMR Amazon; namun, Anda mungkin perlu mengonfigurasi terjemahan alamat jaringan (NAT) dan gateway VPN untuk mengakses layanan atau sumber daya yang berada di luar VPC, misalnya di intranet perusahaan atau titik akhir layanan publik seperti. AWS AWS Key Management Service

> **Important**
>
> Amazon EMR hanya mendukung peluncuran klaster di subnet pribadi dalam versi rilis 4.2 dan yang lebih baru.

Untuk informasi selengkapnya tentang Amazon VPC, lihat Panduan Pengguna Amazon VPC.

Topik

- Opsi Amazon VPC
- Menyiapkan VPC ke klaster host

- Luncurkan klaster ke VPC
- Kebijakan Amazon S3 minimum untuk subnet privat
- Lebih banyak sumber daya untuk mempelajari VPC

Opsi Amazon VPC

Saat Anda meluncurkan klaster Amazon EMR dalam VPC, Anda dapat meluncurkannya baik dalam subnet publik, privat, atau bersama. Terdapat sedikit perbedaan dalam konfigurasi, tergantung pada jenis subnet yang Anda pilih untuk klaster.

Subnet publik

klaster EMR di subnet publik memerlukan gateway internet yang terhubung. Hal ini karena klaster Amazon EMR harus mengakses layanan AWS dan Amazon EMR. Jika layanan, seperti Amazon S3, menyediakan kemampuan untuk membuat VPC endpoint, Anda dapat mengakses layanan tersebut dengan menggunakan titik akhir alih-alih mengakses titik akhir publik melalui gateway internet. Selain itu, Amazon EMR tidak dapat berkomunikasi dengan klaster di subnet publik melalui perangkat terjemahan alamat jaringan (NAT). Gateway internet diperlukan untuk tujuan ini, tetapi Anda masih dapat menggunakan instans NAT atau gateway untuk lalu lintas lain dalam skenario yang lebih kompleks.

Semua instans dalam klaster terhubung ke Amazon S3 melalui VPC endpoint atau gateway internet. Layanan AWS lain yang saat ini tidak mendukung VPC endpoint hanya menggunakan gateway internet.

Jika Anda memiliki sumber daya AWS tambahan yang tidak ingin Anda sambungkan ke gateway internet, Anda dapat meluncurkan komponen tersebut dalam subnet privat yang Anda buat di dalam VPC.

Cluster yang berjalan di subnet publik menggunakan dua grup keamanan: satu untuk node utama dan satu lagi untuk node inti dan tugas. Untuk informasi selengkapnya, lihat Mengendalikan lalu lintas jaringan dengan grup keamanan.

Diagram berikut menunjukkan bagaimana klaster Amazon EMR berjalan di VPC menggunakan subnet publik. Klaster ini dapat terhubung ke sumber daya AWS lain, seperti bucket Amazon S3, melalui gateway internet.

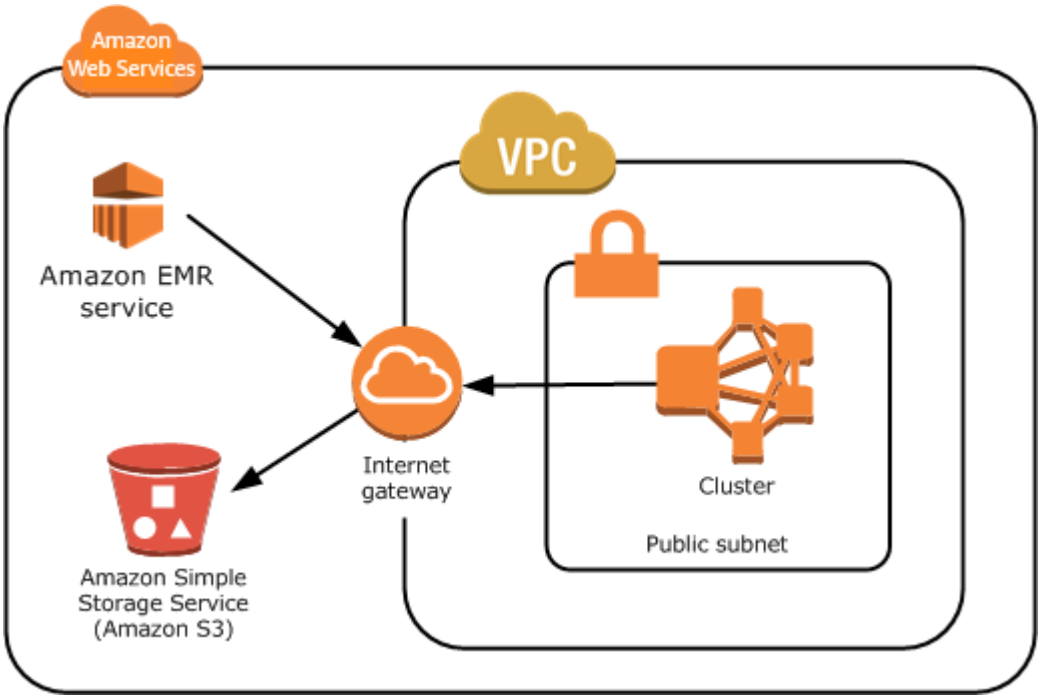
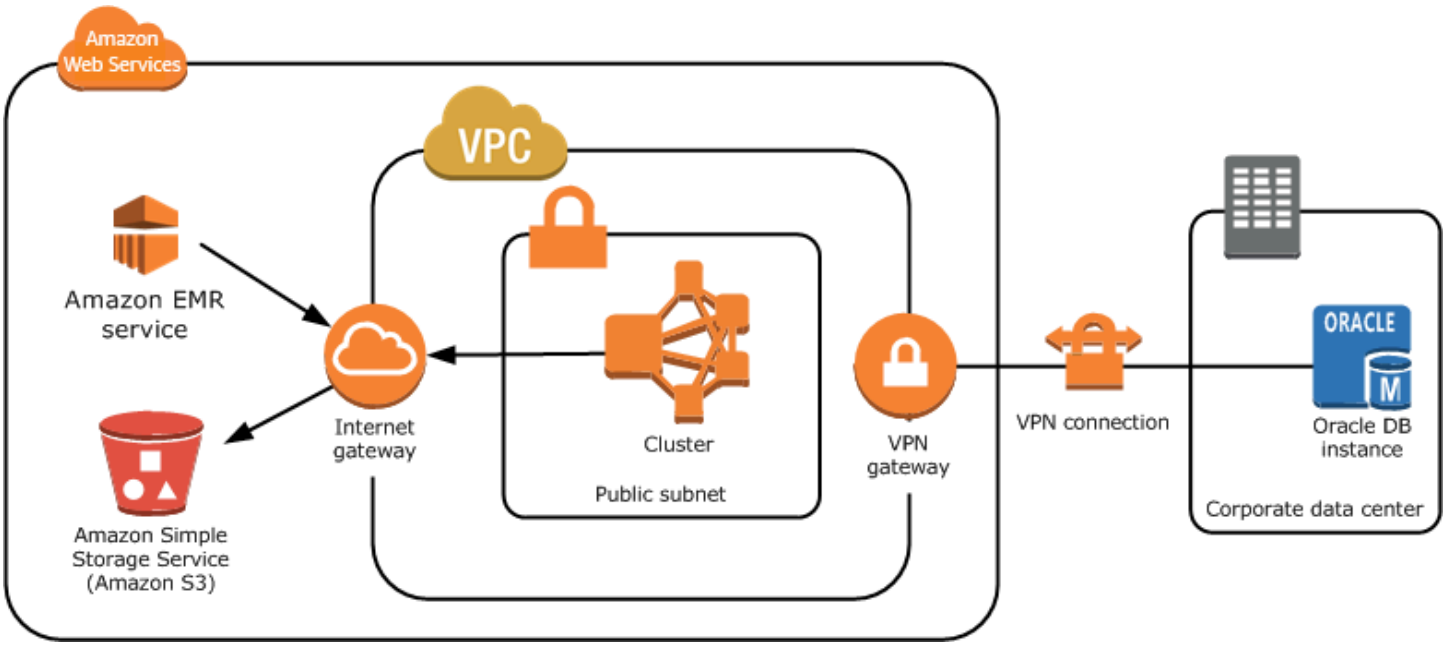


Diagram berikut menunjukkan cara menyiapkan VPC sehingga klaster di VPC dapat mengakses sumber daya di jaringan Anda sendiri, seperti basis data Oracle.



Subnet privat

Subnet pribadi memungkinkan Anda meluncurkan AWS sumber daya tanpa memerlukan subnet untuk memiliki gateway internet terlampir. Amazon EMR mendukung peluncuran cluster di subnet pribadi dengan versi rilis 4.2.0 atau yang lebih baru.

> **Note**
>
> Saat menyiapkan kluster EMR Amazon di subnet pribadi, sebaiknya Anda juga menyiapkan titik akhir VPC untuk Amazon S3. Jika kluster EMR Anda berada dalam subnet pribadi tanpa titik akhir VPC untuk Amazon S3, Anda akan dikenakan biaya gateway NAT tambahan yang terkait dengan lalu lintas S3 karena lalu lintas antara kluster EMR Anda dan S3 tidak akan tetap berada dalam VPC Anda.

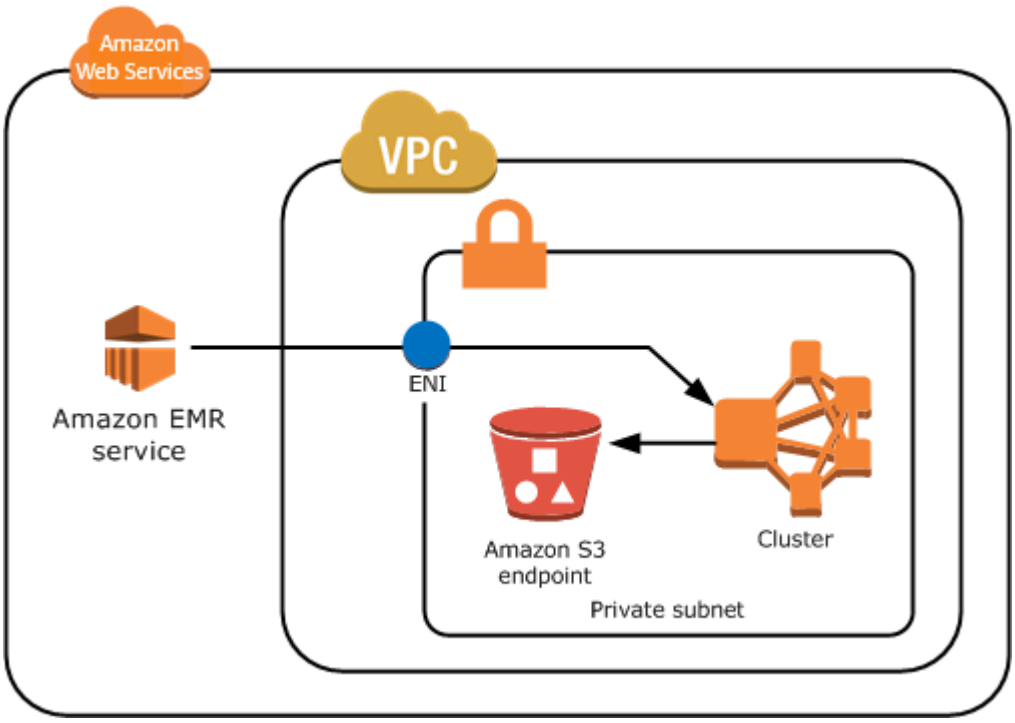
Subnet pribadi berbeda dari subnet publik dengan cara berikut:

- Untuk mengakses layanan AWS yang tidak menyediakan VPC endpoint, Anda tetap harus menggunakan instans NAT atau gateway internet.
- Minimal, Anda harus menyediakan rute ke bucket log layanan Amazon EMR dan repositori Amazon Linux di Amazon S3. Untuk informasi selengkapnya, lihat Kebijakan Amazon S3 minimum untuk subnet privat
- Jika Anda menggunakan fitur EMRFS, Anda harus memiliki VPC endpoint Amazon S3 dan rute dari subnet privat ke DynamoDB.
- Debugging hanya berfungsi jika Anda menyediakan rute dari subnet privat ke titik akhir Amazon SQS publik.
- Membuat konfigurasi subnet privat menggunakan instans NAT atau gateway di subnet publik hanya didukung dengan menggunakan AWS Management Console. Cara termudah untuk menambahkan dan mengonfigurasi instans NAT dan titik akhir VPC Amazon S3 untuk kluster EMR Amazon adalah dengan menggunakan halaman Daftar Subnet VPC di konsol EMR Amazon. Untuk mengonfigurasi gateway NAT, lihat Gateway NAT di Panduan Pengguna Amazon VPC.
- Anda tidak dapat mengubah subnet dengan kluster EMR Amazon yang ada dari publik ke pribadi atau sebaliknya. Untuk menemukan kluster EMR Amazon dalam subnet pribadi, cluster harus dimulai di subnet pribadi itu.

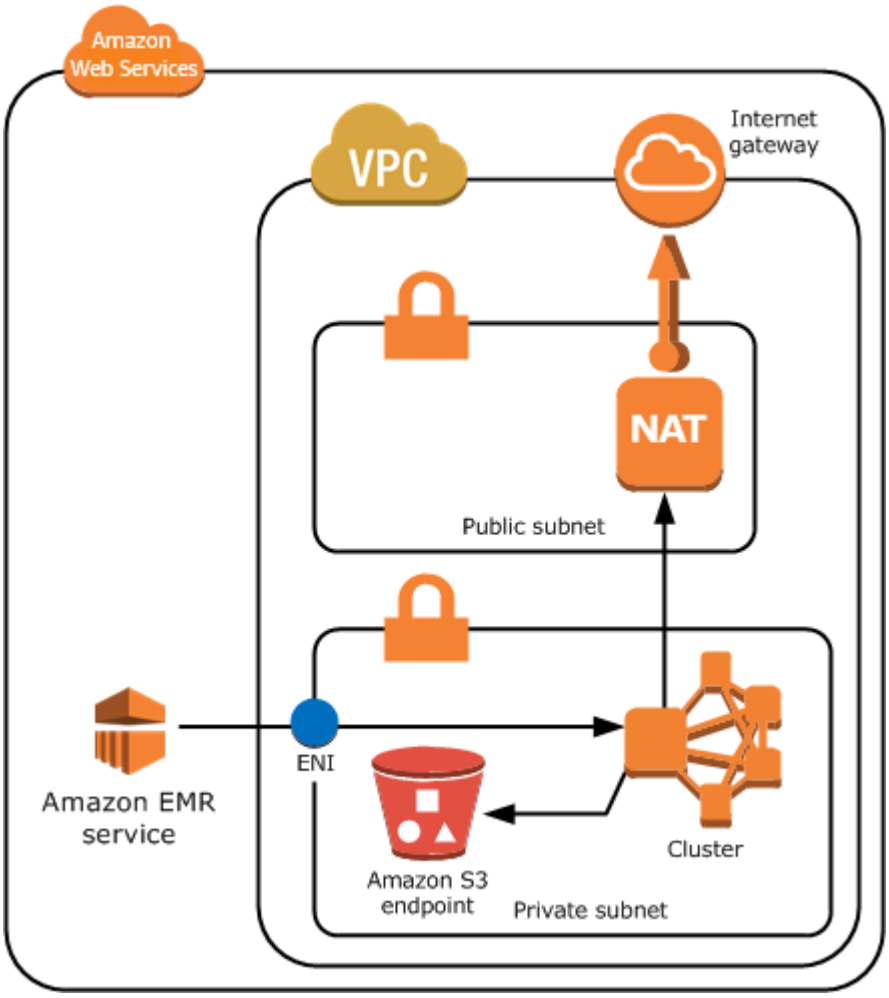
Amazon EMR membuat dan menggunakan grup keamanan default yang berbeda untuk cluster di subnet pribadi: ElasticMapReduce -Master-Private, -Slave-Private, dan -. ElasticMapReduce ElasticMapReduce ServiceAccess Untuk informasi selengkapnya, lihat Mengendalikan lalu lintas jaringan dengan grup keamanan.

Untuk daftar lengkap NACL klaster Anda, pilih Grup keamanan untuk grup Primer dan Keamanan untuk Inti & Tugas di halaman Detail Kluster konsol EMR Amazon.

Gambar berikut menunjukkan bagaimana cluster EMR Amazon dikonfigurasi dalam subnet pribadi. Satu-satunya komunikasi di luar subnet adalah ke Amazon EMR.



Gambar berikut menunjukkan konfigurasi sampel untuk kluster EMR Amazon dalam subnet pribadi yang terhubung ke instance NAT yang berada di subnet publik.



Subnet bersama

Berbagi VPC memungkinkan pelanggan berbagi subnet dengan akun AWS lain dalam AWS Organisasi yang sama. Anda dapat meluncurkan klaster Amazon EMR ke dalam subnet bersama publik dan bersama privat, dengan peringatan berikut.

Pemilik subnet harus berbagi subnet dengan Anda sebelum Anda dapat meluncurkan klaster Amazon EMR ke dalamnya. Namun, subnet yang dibagikan nantinya dapat dibatalkan pembagiannya. Untuk informasi lebih lanjut, lihat Bekerja dengan VPC Bersama. Saat klaster diluncurkan ke subnet bersama dan subnet bersama tersebut kemudian tidak dibatalkan pembagiannya, Anda dapat mengamati perilaku tertentu berdasarkan status klaster Amazon EMR saat subnet tidak dibagikan.

- Subnet dibatalkan pembagiannya sebelum klaster berhasil diluncurkan - Jika pemilik berhenti membagikan Amazon VPC atau subnet saat peserta meluncurkan klaster, klaster akan mengalami gagal memulai atau diinisialisasi sebagian tanpa menyediakan semua instans yang diminta.

- Subnet dibatalkan pembagiannya Setelah klaster berhasil diluncurkan - Saat pemilik berhenti membagikan subnet atau Amazon VPC dengan peserta, klaster peserta tidak akan dapat mengubah ukuran untuk menambahkan instans baru atau mengganti instans yang tidak sehat.

Saat Anda meluncurkan klaster Amazon EMR, beberapa grup keamanan akan dibuat. Dalam subnet bersama, peserta subnet mengontrol grup keamanan ini. Pemilik subnet dapat melihat grup keamanan ini tetapi tidak dapat melakukan tindakan apa pun terhadapnya. Jika pemilik subnet ingin menghapus atau mengubah grup keamanan, peserta yang membuat grup keamanan harus mengambil tindakan.

Kontrol izin VPC dengan IAM

Secara default, semua pengguna dapat melihat semua subnet untuk akun, dan setiap pengguna dapat meluncurkan cluster di subnet apa pun.

Saat meluncurkan cluster ke VPC, Anda dapat menggunakan AWS Identity and Access Management (IAM) untuk mengontrol akses ke cluster dan membatasi tindakan menggunakan kebijakan, seperti yang Anda lakukan dengan cluster yang diluncurkan ke Amazon EC2 Classic. Untuk informasi lebih lanjut tentang IAM, lihat Panduan Pengguna IAM.

Anda juga dapat menggunakan IAM untuk mengontrol siapa yang dapat membuat dan mengelola subnet. Misalnya, Anda dapat membuat satu akun untuk mengelola subnet, dan akun kedua yang dapat meluncurkan cluster tetapi tidak dapat mengubah pengaturan Amazon VPC. Untuk informasi selengkapnya tentang mengelola kebijakan dan tindakan di Amazon EC2 dan Amazon VPC, lihat Kebijakan IAM untuk Amazon EC2 di Panduan Pengguna Amazon EC2 untuk Instans Linux.

Menyiapkan VPC ke klaster host

Sebelum dapat meluncurkan klaster di VPC, Anda harus membuat VPC dan subnet. Untuk subnet publik, Anda harus membuat gateway internet dan melampirkannya ke subnet. Petunjuk berikut menjelaskan cara membuat VPC yang mampu menghosting klaster Amazon EMR.

Untuk membuat VPC dengan subnet untuk klaster Amazon EMR

1. Buka konsol Amazon VPC di https://console.aws.amazon.com/vpc/.
2. Di kanan atas halaman, pilih Wilayah AWSuntuk VPC Anda.
3. Pilih Buat VPC.
4. Pada halaman pengaturan VPC, pilih VPC dan lainnya.

5. Di bawah Generasi otomatis tag nama, aktifkan Generasi otomatis dan masukkan nama untuk VPC Anda. Ini membantu Anda dalam mengidentifikasi VPC dan subnet di konsol Amazon VPC setelah Anda membuatnya.
6. Di bidang blok IPv4 CIDR, masukkan ruang alamat IP pribadi untuk VPC Anda untuk memastikan resolusi nama host DNS yang tepat; jika tidak, Anda mungkin mengalami kegagalan klaster EMR Amazon. Ini termasuk rentang alamat IP berikut:
   - 10.0.0.0 - 10.255.255.255
   - 172.16.0.0 - 172.31.255.255
   - 192.168.0.0 - 192.168.255.255
7. Di bawah Jumlah Availability Zones (AZ), pilih jumlah Availability Zone yang ingin Anda luncurkan subnet.
8. Di bawah Jumlah subnet publik, pilih satu subnet publik untuk ditambahkan ke VPC Anda. Jika data yang digunakan oleh cluster tersedia di internet (misalnya, di Amazon S3 atau Amazon RDS), Anda hanya perlu menggunakan subnet publik dan tidak perlu menambahkan subnet pribadi.
9. Di bawah Jumlah subnet pribadi, pilih jumlah subnet pribadi yang ingin Anda tambahkan ke VPC Anda. Pilih satu atau lebih jika data untuk aplikasi Anda disimpan di jaringan Anda sendiri (misalnya, dalam database Oracle). Untuk VPC di subnet pribadi, semua instans Amazon EC2 minimal harus memiliki rute ke Amazon EMR melalui elastic network interface. Di konsol, hal ini akan secara otomatis dikonfigurasi untuk Anda.
10. Di bawah gateway NAT, secara opsional memilih untuk menambahkan gateway NAT. Mereka hanya diperlukan jika Anda memiliki subnet pribadi yang perlu berkomunikasi dengan internet.
11. Di bawah titik akhir VPC, secara opsional pilih untuk menambahkan titik akhir Amazon S3 ke subnet Anda.
12. Verifikasi bahwa Aktifkan nama host DNS dan Aktifkan resolusi DNS dicentang. Untuk informasi selengkapnya, lihat Menggunakan DNS dengan VPC Anda.
13. Pilih Buat VPC.
14. Jendela status menunjukkan pekerjaan yang sedang berlangsung. Ketika pekerjaan selesai, pilih Lihat VPC untuk menavigasi ke halaman VPC Anda, yang menampilkan VPC default Anda dan VPC yang baru saja Anda buat. VPC yang Anda buat adalah VPC nondefault, oleh karena itu kolom Default VPC menampilkan No.
15. Jika Anda ingin mengaitkan VPC Anda dengan entri DNS yang tidak menyertakan nama domain, navigasikan ke set opsi DHCP, pilih Buat set opsi DHCP, dan hilangkan nama domain. Setelah

Anda membuat set opsi, navigasikan ke VPC baru Anda, pilih Edit opsi DHCP yang diatur di bawah menu Tindakan, dan pilih set opsi baru. Anda tidak dapat mengedit nama domain menggunakan konsol setelah rangkaian opsi DNS dibuat.

Ini adalah praktik terbaik dengan Hadoop dan aplikasi terkait untuk memastikan resolusi nama domain yang sepenuhnya memenuhi syarat (FQDN) untuk simpul. Untuk memastikan resolusi DNS yang tepat, konfigurasikan VPC yang menyertakan set opsi DHCP yang parameternya ditetapkan ke nilai berikut:

- nama domain = **`ec2.internal`**

  Gunakan **`ec2.internal`** jika wilayah Anda adalah US East (N. Virginia). Untuk wilayah lain, gunakan *`nama wilayah`***`.compute.internal`**. Untuk contoh dalam `us-west-2`, gunakan **`us-west-2.compute.internal`**. Untuk Wilayah AWS GovCloud (AS-Barat), gunakan**`us-gov-west-1.compute.internal`**.
- domain-name-servers = **`AmazonProvidedDNS`**

Untuk informasi selengkapnya, lihat Set opsi DHCP di Panduan Pengguna Amazon VPC.

16. Setelah VPC dibuat, buka halaman Subnet dan catat ID Subnet dari salah satu subnet VPC baru Anda. Anda menggunakan informasi ini saat meluncurkan cluster EMR Amazon ke dalam VPC.

Luncurkan klaster ke VPC

Setelah Anda memiliki subnet yang dikonfigurasi untuk menghosting klaster Amazon EMR, luncurkan klaster di subnet tersebut dengan menetapkan pengenal subnet terkait saat membuat klaster.

> **Note**
>
> Amazon EMR mendukung subnet privat dalam versi rilis 4.2 dan di atasnya.

Saat klaster diluncurkan, Amazon EMR menambahkan grup keamanan yang didasarkan pada apakah klaster diluncurkan ke subnet privat atau publik VPC. Semua grup keamanan mengizinkan ingress pada port 8443 guna berkomunikasi ke layanan Amazon EMR, tetapi rentang alamat IP berbeda-beda untuk subnet publik dan privat. Amazon EMR mengelola semua grup keamanan ini, dan mungkin perlu menambahkan alamat IP tambahan ke AWS rentang dari waktu ke waktu. Untuk informasi selengkapnya, lihat Mengendalikan lalu lintas jaringan dengan grup keamanan.

Untuk mengelola cluster pada VPC, Amazon EMR melampirkan perangkat jaringan ke node utama dan mengelolanya melalui perangkat ini. Anda dapat melihat perangkat ini dengan menggunakan tindakan API Amazon EC2 `DescribeInstances`. Jika Anda mengubah perangkat ini dengan cara apapun, klaster dapat mengalami kegagalan.

> **Note**
>
> Kami telah mendesain ulang konsol EMR Amazon agar lebih mudah digunakan. Lihat Apa yang baru dengan konsol? untuk mempelajari tentang perbedaan antara pengalaman konsol lama dan baru.

New console

Untuk meluncurkan cluster ke VPC dengan konsol baru

1. Masuk keAWS Management Console, dan buka konsol EMR Amazon di https://console.aws.amazon.com/emr.
2. Di bawah EMR pada EC2 di panel navigasi kiri, pilih Clusters, lalu pilih Create cluster.
3. Di bawah Networking, buka bidang Virtual Private Cloud (VPC). Masukkan nama VPC Anda atau pilih Browse untuk memilih VPC Anda. Atau, pilih Buat VPC untuk membuat VPC yang dapat Anda gunakan untuk cluster Anda.
4. Pilih opsi lain yang berlaku untuk cluster Anda.
5. Untuk meluncurkan klaster Anda, pilih Buat klaster.

Old console

Untuk meluncurkan cluster ke VPC dengan konsol lama

1. Arahkan ke konsol EMR Amazon baru dan pilih Beralih ke konsol lama dari navigasi samping. Untuk informasi selengkapnya tentang apa yang diharapkan saat beralih ke konsol lama, lihat Menggunakan konsol lama.
2. Pilih Buat klaster.
3. Pilih Pergi ke opsi lanjutan.
4. Di bagian Konfigurasi perangkat keras, untuk Jaringan, pilih ID jaringan VPC yang telah Anda buat sebelumnya.

5. Untuk Subnet EC2, pilih ID subnet yang telah Anda buat sebelumnya.

   a. Jika subnet privat Anda dikonfigurasi sebagaimana mestinya dengan opsi instans NAT dan titik akhir S3, ia akan menampilkan (EMR Ready) di atas nama subnet dan pengenal.

   b. Jika subnet privat Anda tidak memiliki instans NAT dan/atau titik akhir S3, Anda dapat mengonfigurasinya dengan memilih Tambahkan titik akhir S3 dan instans NAT, Tambahkan titik akhir S3, atau Tambahkan instans NAT. Pilih opsi yang diinginkan untuk instans NAT dan titik akhir S3 Anda lalu pilih Konfigurasi.

   > **Important**
   >
   > Untuk membuat instance NAT dari Amazon EMR, Anda memerlukan ec2CreateRoute:,, `ec2:RevokeSecurityGroupEgressec2:AuthorizeSecurityGroupEgress`, `cloudformation:DescribeStackEvents` dan izin. `cloudformation:CreateStack`

   > **Note**
   >
   > Ada biaya tambahan untuk meluncurkan instans Amazon EC2 untuk perangkat NAT Anda.

6. Lanjutkan dengan membuat klaster.

AWS CLI

Untuk meluncurkan cluster ke VPC dengan AWS CLI

> **Note**
>
> AWS CLI tidak menyediakan cara untuk membuat instans NAT secara otomatis dan menghubungkannya ke subnet privat Anda. Namun, untuk membuat titik akhir S3 di subnet Anda, Anda dapat menggunakan perintah Amazon VPC CLI. Gunakan konsol untuk membuat instans NAT dan meluncurkan klaster di subnet privat.

Setelah VPC Anda dikonfigurasi, Anda dapat meluncurkan klaster EMR Amazon di dalamnya dengan menggunakan subperintah dengan parameter. `create-cluster --ec2-attributes` Gunakan parameter `--ec2-attributes` untuk menentukan subnet VPC yang digunakan untuk klaster Anda.

- Untuk membuat cluster di subnet tertentu, ketik perintah berikut, ganti *MyKey dengan nama key* pair Amazon EC2 Anda, *dan* ganti 77XXXX03 dengan subnet ID Anda.

```
aws emr create-cluster --name "Test cluster" --release-label emr-4.2.0 --
applications Name=Hadoop Name=Hive Name=Pig --use-default-roles --ec2-attributes
 KeyName=myKey,SubnetId=subnet-77XXXX03 --instance-type m5.xlarge --instance-
count 3
```

Saat Anda menentukan jumlah instance tanpa menggunakan `--instance-groups` parameter, satu node primer diluncurkan, dan instance yang tersisa diluncurkan sebagai node inti. Semua simpul menggunakan tipe instans yang ditentukan dalam perintah.

> **Note**
>
> Jika Anda belum pernah membuat peran layanan Amazon EMR default dan profil instans EC2, ketik `aws emr create-default-roles` untuk membuatnya sebelum mengetik `create-cluster` subperintah.

Kebijakan Amazon S3 minimum untuk subnet privat

Untuk subnet privat, setidaknya Anda harus menyediakan kemampuan bagi Amazon EMR agar dapat mengakses repositori Amazon Linux. Kebijakan subnet privat ini adalah bagian dari kebijakan VPC endpoint untuk mengakses Amazon S3. Dengan Amazon EMR 5.25.0 atau lebih baru, untuk mengaktifkan akses sekali klik ke server riwayat Spark persisten, Anda harus mengizinkan Amazon EMR untuk mengakses bucket sistem yang mengumpulkan log peristiwa Spark. Jika Anda mengaktifkan pencatatan log, berikan izin PUT ke `aws157-logs-*` bucket. Untuk informasi selengkapnya, lihat Akses sekali klik ke Spark Server Riwayat persisten.

Anda dapat menentukan batasan kebijakan yang memenuhi kebutuhan bisnis sesuai keinginan Anda. Misalnya, Anda dapat menentukan Wilayah `packages.us-east-1.amazonaws.com` untuk menghindari nama bucket Amazon S3 yang ambigu. Contoh kebijakan berikut memberikan izin untuk mengakses repositori Amazon Linux dan bucket sistem Amazon EMR untuk mengumpulkan log

peristiwa Spark. Ganti *MyRegion*dengan Wilayah tempat bucket log Anda berada, misalnya. `us-east-1`

Untuk informasi selengkapnya tentang penggunaan kebijakan IAM dengan titik akhir Amazon VPC, lihat Kebijakan Titik Akhir untuk Amazon S3.

```
{
   "Version": "2008-10-17",
   "Statement": [
       {
           "Sid": "AmazonLinuxAMIRepositoryAccess",
           "Effect": "Allow",
           "Principal": "*",
           "Action": "s3:GetObject",
           "Resource": [
            "arn:aws:s3:::packages.MyRegion.amazonaws.com/*",
            "arn:aws:s3:::repo.MyRegion.amazonaws.com/*",
            "arn:aws:s3:::repo.MyRegion.emr.amazonaws.com/*"
           ]
       },
       {
           "Sid": "EnableApplicationHistory",
           "Effect": "Allow",
           "Principal": "*",
           "Action": [
               "s3:Put*",
               "s3:Get*",
               "s3:Create*",
               "s3:Abort*",
               "s3:List*"
           ],
           "Resource": [
            "arn:aws:s3:::prod.MyRegion.appinfo.src/*"
           ]
       }
   ]
}
```

Contoh kebijakan berikut memberikan izin yang diperlukan untuk mengakses repositori Amazon Linux 2. AMI Amazon Linux 2 adalah default.

```
{
   "Statement": [
```

```
        {
            "Sid": "AmazonLinux2AMIRepositoryAccess",
            "Effect": "Allow",
            "Principal": "*",
            "Action": "s3:GetObject",
            "Resource": [
             "arn:aws:s3:::amazonlinux.MyRegion.amazonaws.com/*",
             "arn:aws:s3:::amazonlinux-2-repos-MyRegion/*"
            ]
        }
    ]
}
```

Lebih banyak sumber daya untuk mempelajari VPC

Gunakan topik berikut untuk mempelajari lebih lanjut tentang VPC dan subnet.

- Subnet Privat dalam VPC
  - Skenario 2: VPC dengan Subnet Publik dan Pribadi (NAT)
  - Instans NAT
  - Ketersediaan Tinggi untuk Instans NAT VPC Amazon: Contoh
- Subnet Publik dalam VPC
  - Skenario 1: VPC dengan Subnet Publik Tunggal
- Informasi VPC Umum
  - Panduan Pengguna Amazon VPC
  - Pengintip VPC
  - Menggunakan Antarmuka Jaringan Elastis dengan VPC Anda
  - Terhubung dengan aman ke instance Linux yang berjalan di VPC pribadi

Membuat sebuah klaster dengan armada instan atau grup instans seragam

Saat Anda membuat cluster dan menentukan konfigurasi node utama, node inti, dan node tugas, Anda memiliki dua opsi konfigurasi. Anda dapat menggunakan Armada instans atau grup instans seragam. Opsi konfigurasi yang Anda pilih berlaku untuk semua simpul, ini berlaku selama masa pakai klaster, dan armada instans serta grup instans tidak dapat bisa berada dalam satu klaster secara bersamaan. Konfigurasi armada instans tersedia di Amazon EMR versi 4.8.0 dan yang lebih baru, tidak termasuk versi 5.0.x.

Anda dapat menggunakan konsol EMR Amazon, API EMR AmazonAWS CLI, atau Amazon EMR untuk membuat cluster dengan konfigurasi mana pun. Saat Anda menggunakan perintah `create-cluster` dari AWS CLI, Anda menggunakan parameter `--instance-fleets` untuk membuat klaster menggunakan armada instans atau, sebagai alternatif, Anda dapat menggunakan parameter `--instance-groups` untuk membuatnya dengan menggunakan grup instans seragam.

Hal yang sama berlaku dengan menggunakan Amazon EMR API. Anda menggunakan konfigurasi `InstanceGroups` untuk menentukan susunan objek `InstanceGroupConfig`, atau Anda menggunakan konfigurasi `InstanceFleets` untuk menentukan susunan objek `InstanceFleetConfig`.

Di konsol EMR Amazon yang baru, Anda dapat memilih untuk menggunakan grup instans atau armada instans saat membuat klaster, dan Anda memiliki opsi untuk menggunakan Instans Spot dengan masing-masing. Dengan konsol EMR Amazon lama, jika Anda menggunakan setelan Opsi Cepat default saat membuat klaster, Amazon EMR menerapkan konfigurasi grup instans seragam ke cluster dan menggunakan Instans Sesuai Permintaan. Untuk menggunakan Instans Spot dengan grup instans seragam, atau untuk mengonfigurasi armada instans dan penyesuaian lainnya, pilih Opsi lanjutan.

Armada instans

Konfigurasi armada instans menawarkan berbagai opsi penyediaan terluas untuk instans Amazon EC2. Setiap jenis simpul memiliki satu armada instans, dan penggunaan armada instans tugas bersifat opsional. Anda dapat menentukan hingga lima jenis instans EC2 per armada, atau 30 jenis instans EC2 per armada saat Anda membuat klaster menggunakan atau AWS CLI Amazon EMR API dan strategi alokasi untuk Instans Sesuai Permintaan dan Spot. Untuk armada instans inti dan tugas, tetapkan Kapasitas target Instans Sesuai Permintaan, dan Instans Spot lainnya. Amazon EMR memilih gabungan apa pun dari tiap instans yang ditentukan untuk memenuhi kapasitas target, menyediakan Instans Sesuai Permintaan dan Spot.

Untuk tipe node utama, Amazon EMR memilih satu jenis instans dari daftar instans Anda, dan Anda menentukan apakah itu disediakan sebagai Instans Sesuai Permintaan atau Spot. Armada instans juga menyediakan opsi tambahan untuk pembelian Instans Spot dan Sesuai Permintaan. Opsi Instans Spot mencakup batas waktu yang menentukan tindakan yang harus diambil jika kapasitas Spot tidak dapat disediakan, dan strategi alokasi pilihan (dioptimalkan kapasitas) untuk meluncurkan armada Instans Spot. Armada Instans Berdasarkan Permintaan juga dapat diluncurkan menggunakan opsi strategi alokasi (harga terendah). Jika Anda menggunakan peran layanan yang bukan merupakan peran layanan default EMR, atau menggunakan kebijakan terkelola EMR dalam

peran layanan, Anda harus menambahkan izin tambahan ke peran layanan klaster kustom untuk mengaktifkan opsi strategi alokasi. Untuk informasi selengkapnya, lihat Peran layanan untuk Amazon EMR (peran EMR).

Untuk informasi lebih lanjut tentang konfigurasi armada instans, lihat Mengkonfigurasi armada instans.

Grup instans seragam

Grup instans seragam menawarkan penyiapan yang lebih sederhana daripada armada instans. Setiap klaster EMR Amazon dapat menyertakan hingga 50 grup instans: satu grup instans utama yang berisi satu instans Amazon EC2, grup instans inti yang berisi satu atau beberapa instans EC2, dan hingga 48 grup instans tugas opsional. Setiap grup instans inti dan tugas dapat berisi sejumlah instans Amazon EC2. Anda dapat menskalakan setiap grup instans dengan menambahkan dan menghapus instans Amazon EC2 secara manual, atau Anda dapat mengatur penskalaan otomatis. Untuk informasi tentang menambahkan dan menghapus instans, lihat Gunakan penskalaan cluster.

Untuk informasi selengkapnya tentang mengonfigurasi grup instans seragam, lihat Konfigurasi grup instans seragam.

Bekerja dengan armada instance dan grup instance

Topik

- Mengkonfigurasi armada instans
- Gunakan pencadangan kapasitas dengan armada instans
- Konfigurasi grup instans seragam
- Praktik terbaik misalnya dan fleksibilitas Availability Zone
- Praktik terbaik untuk konfigurasi klaster

Mengkonfigurasi armada instans

> **Note**
>
> Konfigurasi armada instance hanya tersedia di Amazon EMR rilis 4.8.0 dan yang lebih baru, tidak termasuk 5.0.0 dan 5.0.3.

Konfigurasi armada instans untuk klaster Amazon EMR memungkinkan Anda memilih berbagai macam opsi penyediaan untuk instans Amazon EC2, dan membantu Anda dalam mengembangkan strategi sumber daya yang fleksibel dan elastis untuk setiap jenis simpul di klaster Anda.

Dalam konfigurasi armada instans, tentukan Kapasitas targetuntuk Instans Sesuai Permintaan dan Instans Spot dalam setiap armada. Saat klaster diluncurkan, Amazon EMR menyediakan instans hingga target terpenuhi. Saat Amazon EC2 merebut kembali Instans Spot di klaster yang sedang berjalan karena kenaikan harga atau kegagalan instans, Amazon EMR mencoba mengganti instance dengan jenis instans apa pun yang Anda tentukan. Hal ini memudahkan untuk mendapatkan kembali kapasitas selama lonjakan harga Spot.

Anda dapat menentukan maksimal lima jenis instans Amazon EC2 per armada untuk Amazon EMR untuk digunakan saat memenuhi target, atau maksimal 30 jenis instans Amazon EC2 per armada saat Anda membuat klaster menggunakan atau AWS CLI Amazon EMR API dan strategi alokasi untuk Instans Sesuai Permintaan dan Spot.

Anda juga dapat memilih beberapa subnet untuk Availability Zone yang berbeda. Saat Amazon EMR meluncurkan klaster, ia mencari di seluruh subnet tersebut guna menemukan instans dan opsi pembelian yang Anda tentukan. Jika Amazon EMR mendeteksi peristiwa AWS skala besar di satu atau beberapa Availability Zone, Amazon EMR secara otomatis mencoba merutekan lalu lintas dari Availability Zone yang terkena dampak dan mencoba meluncurkan cluster baru yang Anda buat di Availability Zone alternatif sesuai dengan pilihan Anda. Perhatikan bahwa pemilihan Zona Ketersediaan klaster hanya terjadi pada pembuatan klaster. Node cluster yang ada tidak secara otomatis diluncurkan kembali di Availability Zone baru jika terjadi pemadaman Availability Zone.

Pertimbangan-pertimbangan

Pertimbangkan item berikut saat Anda menggunakan armada instans dengan Amazon EMR.

- Anda dapat memiliki satu armada instance, dan hanya satu, per tipe node (primer, inti, tugas). Anda dapat menentukan hingga lima jenis instans Amazon EC2 untuk setiap armada di AWS Management Console (atau maksimal 30 jenis per armada instans saat membuat klaster menggunakan atau AWS CLI Amazon EMR API dan file). Strategi alokasi untuk armada instans
- Amazon EMR memilih salah satu atau semua jenis instans Amazon EC2 yang ditentukan untuk disediakan dengan opsi pembelian Spot dan Sesuai Permintaan.
- Anda dapat menetapkan kapasitas target bagi Instans Spot dan Sesuai Permintaan untuk armada inti dan armada tugas. Gunakan vCPU atau unit generik yang ditetapkan ke setiap instans Amazon EC2 yang diperhitungkan terhadap target. Amazon EMR menyediakan instans hingga setiap kapasitas target terpenuhi sepenuhnya. Untuk armada utama, targetnya selalu satu.

- Anda dapat memilih satu subnet (Availability Zone) atau rentang. Jika Anda memilih rentang, Amazon EMR akan menyediakan kapasitas di Availability Zone yang paling sesuai.
- Saat Anda menentukan kapasitas target untuk Instans Spot:
  - Untuk setiap jenis instans, tentukan harga Spot maksimum. Amazon EMR menyediakan Instans Spot jika harga Spot di bawah harga Spot maksimum. Anda tidak selalu membayar harga Spot dengan harga Spot maksimum.
  - Untuk setiap armada, tentukan periode batas waktu untuk menyediakan Instans Spot. Jika Amazon EMR tidak dapat menyediakan kapasitas Spot, Anda dapat mengakhiri klaster atau beralih ke penyediaan kapasitas Sesuai Permintaan sebagai gantinya. Ini hanya berlaku untuk penyediaan cluster, bukan mengubah ukurannya. Jika periode batas waktu berakhir selama proses pengubahan ukuran klaster, permintaan Spot yang tidak tersedia akan dibatalkan tanpa mentransfer ke kapasitas Sesuai Permintaan.
- Untuk setiap armada, Anda dapat menentukan salah satu strategi alokasi berikut untuk Instans Spot Anda: kapasitas harga yang dioptimalkan, dioptimalkan kapasitas, harga terendah, atau terdiversifikasi di semua kumpulan.
- Untuk setiap armada, Anda dapat menerapkan strategi alokasi harga terendah untuk Instans Sesuai Permintaan; Anda tidak dapat menyesuaikan strategi alokasi untuk Instans Sesuai Permintaan.
- Untuk setiap armada dengan Sesuai Permintaan `allocation strategy - lowest-price`, Anda dapat memilih untuk menerapkan opsi pencadangan kapasitas.
- Periksa ukuran subnet Anda sebelum meluncurkan cluster Anda. Saat Anda menyediakan cluster dengan armada tugas dan tidak ada cukup alamat IP yang tersedia di subnet yang sesuai, armada akan masuk ke status ditangguhkan alih-alih mengakhiri cluster dengan kesalahan. Untuk menghindari masalah ini, kami sarankan untuk meningkatkan jumlah alamat IP di subnet Anda.

Opsi armada instans

Gunakan panduan berikut untuk memahami opsi armada instans.

Topik

Menetapkan kapasitas target

Tentukan kapasitas target yang Anda inginkan untuk armada inti dan armada tugas. Saat Anda melakukannya, ia akan menentukan jumlah Instans Sesuai Permintaan dan Instans Spot yang disediakan oleh Amazon EMR. Saat Anda menentukan sebuah instans, Anda memutuskan berapa banyak setiap instans diperhitungkan terhadap target. Saat Instans Sesuai Permintaan disediakan, ia diperhitungkan dalam target Sesuai Permintaan. Hal yang sama berlaku untuk Instans Spot. Tidak seperti armada inti dan tugas, armada utama selalu satu contoh. Oleh karena itu, target kapasitas armada ini selalu satu.

Saat Anda menggunakan konsol, vCPU jenis instans Amazon EC2 digunakan sebagai hitungan kapasitas target secara default. Anda dapat mengubah ini ke Unit umum, lalu tentukan jumlah untuk setiap jenis instans EC2. Saat Anda menggunakan AWS CLI, secara manual tetapkan unit umum untuk setiap tipe instans.

> **Important**
>
> Saat Anda memilih tipe instans menggunakan AWS Management Console, jumlah vCPU yang ditampilkan untuk setiap Tipe instans adalah jumlah vcore YARN untuk tipe instans tersebut, bukan jumlah vCPU EC2 untuk tipe instans tersebut. Untuk informasi selengkapnya tentang jumlah vCPU untuk setiap tipe instans, lihat Tipe Instans Amazon EC2.

Untuk setiap armada, Anda menentukan hingga lima jenis instans Amazon EC2. Jika Anda menggunakan Strategi alokasi untuk armada instans dan membuat klaster menggunakan AWS CLI atau Amazon EMR API, Anda dapat menentukan hingga 30 jenis instans EC2 per armada instans. Amazon EMR memilih kombinasi apa pun dari tipe instans EC2 ini untuk memenuhi kapasitas target Anda. Karena Amazon EMR ingin memenuhi kapasitas target sepenuhnya, kelebihan penggunaan dapat terjadi. Misalnya, jika ada dua unit yang tidak terpenuhi, dan Amazon EMR hanya dapat menyediakan instans dengan jumlah lima unit, instans tersebut masih mendapatkan penyediaan, artinya kapasitas target terlampaui tiga unit.

Jika Anda mengurangi kapasitas target untuk mengubah ukuran klaster yang sedang berjalan, Amazon EMR akan mencoba menyelesaikan tugas aplikasi dan mengakhiri instans untuk memenuhi target baru. Untuk informasi selengkapnya, lihat Akhiri pada penyelesaian tugas.

Opsi peluncuran

Untuk Instans Spot, Anda dapat menentukan Harga Spot maksimum untuk setiap tipe instans dalam armada. Anda dapat menetapkan harga ini sebagai persentase dari harga Sesuai Permintaan, atau sebagai jumlah dolar tertentu. Amazon EMR menyediakan Instans Spot jika harga Spot saat ini di Availability Zone berada di bawah harga Spot maksimum Anda. Anda tidak selalu membayar harga Spot dengan harga Spot maksimum.

> **Note**
>
> Instans Spot dengan durasi yang ditentukan (juga dikenal sebagai blok Spot) tidak lagi tersedia untuk pelanggan baru mulai 1 Juli 2021. Untuk pelanggan yang sebelumnya telah menggunakan fitur ini, kami akan terus mendukung Instans Spot dengan durasi yang ditentukan hingga 31 Desember 2022.

Tersedia di Amazon EMR 5.12.1 dan yang lebih baru, Anda memiliki opsi untuk meluncurkan armada Instans Spot dan Sesuai Permintaan dengan alokasi kapasitas yang dioptimalkan. Opsi strategi alokasi ini dapat diatur dalam yang lama AWS Management Console atau menggunakan API`RunJobFlow`. Perhatikan bahwa Anda tidak dapat menyesuaikan strategi alokasi di konsol baru. Menggunakan opsi strategi alokasi memerlukan izin peran layanan tambahan. Jika Anda menggunakan peran layanan Amazon EMR default dan kebijakan terkelola (`EMR_DefaultRole`dan`AmazonEMRServicePolicy_v2`) untuk klaster, izin untuk opsi strategi alokasi sudah disertakan. Jika Anda tidak menggunakan peran layanan Amazon EMR default dan kebijakan terkelola, Anda harus menambahkannya untuk menggunakan opsi ini. Lihat Peran layanan untuk Amazon EMR (peran EMR).

Untuk informasi selengkapnya tentang Instans Spot, lihat Instans Spot di Panduan Pengguna Amazon EC2 untuk Instans Linux. Untuk informasi selengkapnya tentang Instans Sesuai Permintaan, lihat Instans Sesuai Permintaan di Panduan Pengguna Amazon EC2 untuk Instans Linux.

Jika Anda memilih untuk meluncurkan armada Instans Sesuai Permintaan dengan strategi alokasi harga terendah, Anda memiliki opsi untuk menggunakan pencadangan kapasitas. Opsi pencadangan kapasitas dapat diatur menggunakan API Amazon EMR `RunJobFlow`. Pencadangan kapasitas memerlukan izin peran layanan tambahan yang harus Anda tambahkan untuk menggunakan opsi ini. Lihat Izin strategi alokasi. Perhatikan bahwa Anda tidak dapat menyesuaikan reservasi kapasitas di konsol baru.

Beberapa opsi subnet (Availability Zones)

Saat menggunakan armada instans, Anda dapat menentukan beberapa subnet Amazon EC2 dalam VPC, masing-masing terkait dengan Availability Zone yang berbeda. Jika Anda menggunakan EC2-Classic, Anda menentukan Availability Zone secara eksplisit. Amazon EMR mengidentifikasi Availability Zone terbaik untuk meluncurkan instans sesuai dengan spesifikasi armada Anda. Instans selalu disediakan hanya dalam satu Availability Zone. Anda dapat memilih subnet privat atau subnet publik, tetapi Anda tidak dapat menggabungkan keduanya, dan subnet yang Anda tentukan harus berada dalam VPC yang sama.

Konfigurasi simpul master

Karena armada instance utama hanya satu instance, konfigurasinya sedikit berbeda dari armada instance inti dan tugas. Anda hanya memilih On-Demand atau Spot untuk armada instans utama karena hanya terdiri dari satu instance. Jika Anda menggunakan konsol untuk membuat armada instans, kapasitas target untuk opsi pembelian yang Anda pilih akan disetel ke 1. Jika Anda menggunakan AWS CLI, selalu atur baik `TargetSpotCapacity` atau `TargetOnDemandCapacity` ke 1 yang sesuai. Anda masih dapat memilih hingga lima jenis instans untuk armada instans utama (atau maksimal 30 saat Anda menggunakan opsi strategi alokasi untuk Instans On-Demand atau Spot). Namun, tidak seperti armada instance inti dan tugas, di mana Amazon EMR dapat menyediakan beberapa instance dari berbagai jenis, Amazon EMR memilih satu jenis instans untuk disediakan untuk armada instans utama.

Strategi alokasi untuk armada instans

Dengan Amazon EMR versi 5.12.1 dan yang lebih baru, Anda dapat menggunakan opsi strategi alokasi dengan Instans Sesuai Permintaan dan Spot untuk setiap simpul klaster. Saat membuat klaster menggunakanAWS CLI, Amazon EMR API, atau konsol EMR Amazon dengan strategi alokasi, Anda dapat menentukan hingga 30 jenis instans Amazon EC2 per armada. Dengan konfigurasi armada instans klaster EMR Amazon default, Anda dapat memiliki hingga 5 jenis instans per armada. Kami merekomendasikan Anda untuk menggunakan opsi strategi alokasi untuk penyediaan klaster yang lebih cepat, alokasi Instans Spot yang lebih akurat, dan gangguan Instans Spot yang lebih sedikit.

Topik

- Strategi alokasi dengan Instans Sesuai Permintaan
- Strategi alokasi dengan Instans Spot
- Izin strategi alokasi

- Izin IAM yang diperlukan untuk strategi alokasi

Strategi alokasi dengan Instans Sesuai Permintaan

Ketika Anda menggunakan strategi alokasi, Instans On-Demand Anda menggunakan strategi harga terendah. Ini meluncurkan instance dengan harga terendah terlebih dahulu. Saat meluncurkan Instans Sesuai Permintaan, Anda dapat menggunakan reservasi kapasitas terbuka atau bertarget di akun Anda. Anda dapat menggunakan reservasi kapasitas terbuka untuk node primer, inti, dan tugas. Anda mungkin mengalami kapasitas yang tidak mencukupi dengan Instans Sesuai Permintaan dengan strategi alokasi misalnya armada. Kami menyarankan Anda menentukan jumlah jenis instans yang lebih besar untuk diversifikasi dan mengurangi kemungkinan mengalami kapasitas yang tidak mencukupi. Untuk informasi selengkapnya, lihat Gunakan pencadangan kapasitas dengan armada instans.

Strategi alokasi dengan Instans Spot

Untuk Instans Spot, Anda dapat memilih salah satu strategi alokasi berikut:

`price-capacity-optimized` (direkomendasikan)

Strategi alokasi yang dioptimalkan dengan kapasitas harga meluncurkan instans Spot dari kumpulan instans Spot yang memiliki kapasitas tertinggi yang tersedia dan harga terendah untuk jumlah instans yang diluncurkan. Akibatnya, strategi yang dioptimalkan dengan kapasitas harga biasanya memiliki peluang lebih tinggi untuk mendapatkan kapasitas Spot, dan memberikan tingkat interupsi yang lebih rendah.

`capacity-optimized`

Strategi alokasi yang dioptimalkan kapasitas meluncurkan Instans Spot ke dalam kumpulan yang paling tersedia dengan peluang interupsi terendah dalam waktu dekat. Ini adalah pilihan yang baik untuk beban kerja yang mungkin memiliki biaya interupsi yang lebih tinggi terkait dengan pekerjaan yang dimulai ulang. Ini adalah strategi default untuk Amazon EMR rilis 6.9.0 dan lebih rendah.

`diversified`

Dengan strategi alokasi yang beragam, Amazon EC2 mendistribusikan Instans Spot di semua kumpulan kapasitas Spot.

`lowest-price`

Strategi alokasi harga terendah meluncurkan Instans Spot dari kumpulan harga terendah yang memiliki kapasitas yang tersedia. Jika kolam dengan harga terendah tidak memiliki kapasitas yang tersedia, Instans Spot berasal dari kolam dengan harga terendah berikutnya yang memiliki kapasitas yang tersedia. Jika kolam kehabisan kapasitas sebelum memenuhi kapasitas yang Anda minta, armada Amazon EC2 mengambil dari kolam dengan harga terendah berikutnya untuk terus memenuhi permintaan Anda. Untuk memastikan bahwa kapasitas yang Anda inginkan terpenuhi, Anda mungkin menerima Instans Spot dari beberapa kolam. Karena strategi ini hanya mempertimbangkan harga instans, dan tidak mempertimbangkan ketersediaan kapasitas, hal itu dapat menyebabkan tingkat interupsi yang tinggi.

Izin strategi alokasi

Opsi strategi alokasi memerlukan beberapa izin IAM yang secara otomatis disertakan dalam peran layanan EMR Amazon default dan kebijakan terkelola Amazon EMR (dan). `EMR_DefaultRole` `AmazonEMRServicePolicy_v2` Jika Anda menggunakan peran layanan kustom atau kebijakan terkelola untuk klaster, Anda harus menambahkan izin ini sebelum membuat klaster. Untuk informasi selengkapnya, lihat Izin strategi alokasi.

Pencadangan Kapasitas Sesuai Permintaan (ODCR) opsional tersedia saat Anda menggunakan opsi strategi alokasi Sesuai Permintaan. Opsi pencadangan kapasitas memungkinkan Anda menentukan preferensi dalam menggunakan kapasitas yang dicadangkan terlebih dahulu untuk klaster Amazon EMR. Anda dapat menggunakan ini untuk memastikan bahwa beban kerja kritis Anda menggunakan kapasitas yang telah Anda cadangkan menggunakan ODCR terbuka atau tertarget. Untuk beban kerja yang non-kritis, preferensi pencadangan kapasitas memungkinkan Anda menentukan apakah kapasitas yang dicadangkan harus digunakan.

Pencandangan kapasitas hanya dapat digunakan oleh instans yang cocok dengan atributnya (tipe instans, platform, dan Availability Zone). Secara default, pencadangan kapasitas terbuka akan secara otomatis digunakan oleh Amazon EMR saat menyediakan Instans Sesuai Permintaan yang cocok dengan atribut instans. Jika Anda tidak memiliki instans berjalan yang cocok dengan atribut pencadangan kapasitas, instans tersebut tetap tidak digunakan hingga Anda meluncurkan instans yang cocok dengan atributnya. Jika Anda tidak ingin menggunakan pencadangan kapasitas apa pun saat meluncurkan klaster, Anda harus menyetel preferensi pencadangan kapasitas ke tidak ada dalam opsi peluncuran.

Namun, Anda juga dapat menargetkan pencadangan kapasitas untuk beban kerja tertentu. Ini memungkinkan Anda untuk secara eksplisit mengontrol instans mana yang diizinkan untuk berjalan dalam kapasitas yang dicadangkan itu. Untuk informasi selengkapnya tentang reservasi kapasitas Sesuai Permintaan, lihat. Gunakan pencadangan kapasitas dengan armada instans

Izin IAM yang diperlukan untuk strategi alokasi

Peran layanan untuk Amazon EMR (peran EMR) Anda memerlukan izin tambahan guna membuat cluster yang menggunakan opsi strategi alokasi untuk armada Instans Sesuai Permintaan atau Spot.

Kami secara otomatis menyertakan izin ini dalam `EMR_DefaultRole`peran layanan EMR Amazon default dan kebijakan terkelola Amazon EMR. `AmazonEMRServicePolicy_v2`

Jika Anda menggunakan peran layanan kustom atau kebijakan terkelola untuk klaster, Anda harus menambahkan izin berikut:

```
{
 "Version": "2012-10-17",
 "Statement": [
  {
   "Effect": "Allow",
   "Action": [
    "ec2:DeleteLaunchTemplate",
    "ec2:CreateLaunchTemplate",
    "ec2:DescribeLaunchTemplates",
    "ec2:CreateLaunchTemplateVersion",
    "ec2:CreateFleet"
   ],
   "Resource": "*"
  }
}
```

Izin peran layanan berikut diperlukan untuk membuat klaster yang menggunakan reservasi kapasitas terbuka atau bertarget. Anda harus menyertakan izin ini selain izin yang diperlukan untuk menggunakan opsi strategi alokasi.

Example Dokumen kebijakan untuk pencadangan kapasitas peran layanan

Untuk menggunakan pencadangan kapasitas terbuka, Anda harus menyertakan izin tambahan berikut.

```
{
```

```
    "Version": "2012-10-17",
    "Statement": [
        {
            "Effect": "Allow",
            "Action": [
                "ec2:DescribeCapacityReservations",
                "ec2:DescribeLaunchTemplateVersions",
                "ec2:DeleteLaunchTemplateVersions"
            ],
            "Resource": "*"
        }
    ]
}
```

Example

Untuk menggunakan pencadangan kapasitas yang ditargetkan, Anda harus menyertakan izin tambahan berikut.

```
{
    "Version": "2012-10-17",
    "Statement": [
        {
            "Effect": "Allow",
            "Action": [
                "ec2:DescribeCapacityReservations",
                "ec2:DescribeLaunchTemplateVersions",
                "ec2:DeleteLaunchTemplateVersions",
                "resource-groups:ListGroupResources"
            ],
            "Resource": "*"
        }
    ]
}
```

Konfigurasikan armada instance untuk klaster Anda

> **Note**
>
> Kami telah mendesain ulang konsol EMR Amazon agar lebih mudah digunakan. Lihat Apa yang baru dengan konsol? untuk mempelajari tentang perbedaan antara pengalaman konsol lama dan baru.

New console

Untuk membuat cluster dengan armada instance dengan konsol baru

1. Masuk keAWS Management Console, dan buka konsol EMR Amazon di https://console.aws.amazon.com/emr.
2. Di bawah EMR pada EC2 di panel navigasi kiri, pilih Clusters, dan pilih Create cluster.
3. Di bawah konfigurasi Cluster, pilih Armada Instance.
4. Untuk setiap grup Node, pilih Tambahkan jenis instans dan pilih hingga 5 tipe instans untuk armada instance primer dan inti dan hingga lima belas tipe instans untuk armada instance tugas. Amazon EMR mungkin menyediakan campuran jenis instans ini saat meluncurkan cluster.
5. Di bawah setiap tipe grup node, pilih menu tarik-turun Tindakan di samping setiap instance untuk mengubah pengaturan ini:

   Tambahkan volume EBS

   Tentukan volume EBS yang akan dilampirkan ke jenis instans setelah Amazon EMR menyediakannya.

   Edit kapasitas tertimbang

   Untuk grup node inti, ubah nilai ini ke sejumlah unit yang sesuai dengan aplikasi Anda. Jumlah VCores YARN untuk setiap jenis instance armada digunakan sebagai unit kapasitas tertimbang default. Anda tidak dapat mengedit kapasitas tertimbang untuk node utama.

   Edit harga Spot maksimum

   Tentukan harga Spot maksimum untuk setiap jenis instans dalam armada. Anda dapat menetapkan harga ini sebagai persentase dari harga Sesuai Permintaan, atau sebagai jumlah dolar tertentu. Jika harga Spot saat ini di Availability Zone di bawah harga Spot maksimum Anda, Amazon EMR menyediakan Instans Spot. Anda tidak selalu membayar harga Spot dengan harga Spot maksimum.
6. Secara opsional, untuk menambahkan grup keamanan untuk node Anda, perluas grup keamanan EC2 (firewall) di bagian Jaringan dan pilih grup keamanan Anda untuk setiap jenis node.
7. Secara opsional, pilih kotak centang di samping Terapkan strategi alokasi jika Anda ingin menggunakan opsi strategi alokasi, dan pilih strategi alokasi yang ingin Anda tentukan untuk

Instans Spot. Anda tidak boleh memilih opsi ini jika peran layanan EMR Amazon Anda tidak memiliki izin yang diperlukan. Untuk informasi selengkapnya, lihat Strategi alokasi untuk armada instans.

8. Pilih opsi lain yang berlaku untuk cluster Anda.
9. Untuk meluncurkan klaster Anda, pilih Buat klaster.

Old console

Untuk membuat cluster dengan armada instance dengan konsol lama

1. Arahkan ke konsol EMR Amazon baru dan pilih Beralih ke konsol lama dari navigasi samping. Untuk informasi selengkapnya tentang apa yang diharapkan saat beralih ke konsol lama, lihat Menggunakan konsol lama.
2. Pilih Buat klaster.
3. Di bagian atas jendela konsol, pilih Pergi ke opsi lanjutan, masukkan opsi Konfigurasi perangkat lunak, lalu pilih Selanjutnya.
4. Di bawah Komposisi klaster, pilih Armasa instans. Jika Anda memilih opsi armada instans, Anda akan melihat opsi untuk menentukan Kapasitas target Instans Spot dan Sesuai Permintaan muncul di tabel Simpul Klaster dan Instans.
5. Untuk Jaringan, masukkan nilai. Jika Anda memilih VPC untuk Jaringan, pilih satu Subnet EC2 atau CTRL+klik untuk memilih beberapa subnet Amazon EC2. Subnet yang Anda pilih harus berjenis sama (privat atau pribadi). Jika Anda memilih hanya satu, klaster Anda akan diluncurkan di subnet tersebut. Jika Anda memilih grup, subnet yang paling sesuai akan dipilih dari grup saat klaster diluncurkan.

   > **Note**
   >
   > Akun dan Wilayah Anda dapat memberi Anda opsi untuk memilih Luncurkan ke EC2-Classic untuk Jaringan. Jika Anda memilih opsi tersebut, pilih satu atau lebih dari Availability Zone EC2 bukan Subnet EC2. Untuk informasi selengkapnya, lihat Amazon EC2 dan Amazon VPC di Panduan Pengguna Amazon EC2 untuk Instans Linux.

6. Di bawah Strategi Alokasi, pilih kotak centang untuk menerapkan strategi alokasi jika Anda ingin menggunakan opsi strategi alokasi. Untuk informasi selengkapnya, lihat Strategi alokasi untuk armada instans.

7. Untuk setiap Jenis Simpul, jika Anda ingin mengubah nama default armada instans, pilih ikon pensil, lalu masukkan nama yang familiar. Jika ingin menghapus armada instans Tugas, pilih ikon X di sisi kanan baris Tugas.
8. Pilih Tambah/hapus jenis instans ke armada dan pilih hingga lima jenis instans dari daftar untuk armada instance primer dan inti; tambahkan hingga lima belas jenis instans untuk armada instance tugas. Amazon EMR dapat memilih untuk menyediakan gabungan apa pun dari tipe instans ini saat meluncurkan klaster.
9. Untuk setiap jenis instans inti dan tugas, pilih bagaimana Anda ingin menentukan kapasitas tertimbang (Setiap instans dihitung sebagai unit X) untuk instans tersebut. Jumlah YARN vCore untuk setiap jenis instans Armada digunakan sebagai unit kapasitas tertimbang default, tetapi Anda dapat mengubah nilainya ke unit apa pun yang sesuai untuk aplikasi Anda.
10. Di bawah Kapasitas target, tentukan jumlah total Instans Sesuai Permintaan dan Spot yang Anda inginkan per armada. EMR memastikan bahwa instans dalam armada memenuhi unit yang diminta untuk kapasitas target Sesuai Permintaan dan Spot. Jika tidak ada unit Sesuai Permintaan atau Spot yang ditentukan untuk armada, maka tidak ada kapasitas yang disediakan untuk armada tersebut.
11. Jika armada dikonfigurasi dengan kapasitas Target untuk Spot, Anda dapat memasukkan harga Spot maksimum sebagai % dari harga Sesuai Permintaan, atau Anda dapat memasukkan jumlah Dolar ($) dalam USD.
12. Agar volume EBS dilampirkan ke tipe instans saat disediakan, pilih pensil di sebelah Penyimpanan EBS lalu masukkan opsi konfigurasi EBS.
13. Jika Anda menetapkan jumlah instan untuk Unit spot, setel Opsi Spot Lanjutan sesuai dengan panduan berikut:

    - Batas waktu penyediaan—Gunakan pengaturan ini untuk mengontrol apa yang dilakukan Amazon EMR saat tidak dapat menyediakan Instans Spot dari salah diantara Tipe instans Armada yang Anda tentukan. Masukkan periode batas waktu dalam hitungan menit, lalu pilih apakah akan Mengakhiri klaster atau Beralih ke penyediaan Instans Sesuai Permintaan. Jika Anda memilih untuk beralih ke Instans Sesuai Permintaan, kapasitas Instans Sesuai Permintaan yang ditetapkan akan diperhitungkan dalam kapasitas target untuk Instans Spot, dan Amazon EMR menyediakan Instans Sesuai Permintaan hingga kapasitas target untuk Instans Spot terpenuhi.
14. Pilih Selanjutnya, ubah pengaturan klaster lainnya, lalu pilih Selanjutnya.

15. Jika Anda memilih untuk menerapkan opsi strategi alokasi baru, dalam pengaturan Opsi Keamanan, pilih peran EMR dan profil instans EC2 yang berisi izin yang diperlukan untuk opsi strategi alokasi. Jika tidak, pembuatan klaster akan gagal.
16. Pilih Buat Klaster.

AWS CLI

Untuk membuat dan meluncurkan cluster dengan armada instance denganAWS CLI, ikuti panduan berikut:

- Untuk membuat dan meluncurkan sebuah klaster dengan armada instans, gunakan perintah `create-cluster` bersama dengan parameter `--instance-fleet`.
- Untuk mendapatkan detail konfigurasi tentang armada instance dalam sebuah cluster, gunakan `list-instance-fleets` perintah.
- Untuk menambahkan beberapa AMI Amazon Linux kustom ke cluster yang Anda buat, gunakan `CustomAmiId` opsi dengan setiap `InstanceType` spesifikasi. Anda dapat mengonfigurasi node armada instance dengan beberapa jenis instans dan beberapa AMI kustom agar sesuai dengan kebutuhan Anda. Lihat Contoh: Membuat cluster dengan konfigurasi armada instance.
- Untuk membuat perubahan pada kapasitas target untuk armada instance, gunakan `modify-instance-fleet` perintah.
- Untuk menambahkan armada instance tugas ke cluster yang belum memilikinya, gunakan `add-instance-fleet` perintah.
- Beberapa AMI kustom dapat ditambahkan ke armada instance tugas menggunakan CustomAmiId argumen dengan add-instance-fleet perintah. Lihat Contoh: Membuat cluster dengan konfigurasi armada instance.
- Untuk menggunakan opsi strategi alokasi saat membuat armada instance, perbarui peran layanan untuk menyertakan dokumen kebijakan contoh di bagian berikut.
- Untuk menggunakan opsi reservasi kapasitas saat membuat armada instans dengan strategi alokasi Sesuai Permintaan, perbarui peran layanan untuk menyertakan dokumen kebijakan contoh di bagian berikut.
- Armada instans secara otomatis disertakan dalam peran layanan EMR default dan kebijakan terkelola Amazon EMR (dan). `EMR_DefaultRole` `AmazonEMRServicePolicy_v2` Jika Anda menggunakan peran layanan kustom atau kebijakan terkelola khusus untuk klaster, Anda harus menambahkan izin baru untuk strategi alokasi di bagian berikut.

Contoh: Membuat cluster dengan konfigurasi armada instance

Contoh berikut menunjukkan `create-cluster` perintah dengan berbagai opsi yang dapat Anda gabungkan.

> **Note**
>
> Jika sebelumnya Anda belum membuat peran layanan EMR Amazon default dan profil instans EC2, gunakan `aws emr create-default-roles` untuk membuatnya sebelum menggunakan perintah. `create-cluster`

Example Contoh: Primer On-Demand, inti On-Demand dengan tipe instans tunggal, VPC Default

```
aws emr create-cluster --release-label emr-5.3.1 --service-role EMR_DefaultRole \
  --ec2-attributes InstanceProfile=EMR_EC2_DefaultRole \
  --instance-fleets \

 InstanceFleetType=MASTER,TargetOnDemandCapacity=1,InstanceTypeConfigs=['{InstanceType=m5.xlarg
 \

 InstanceFleetType=CORE,TargetOnDemandCapacity=1,InstanceTypeConfigs=['{InstanceType=m5.xlarge}
```

Example Contoh: Spot primer, inti Spot dengan tipe instance tunggal, VPC default

```
aws emr create-cluster --release-label emr-5.3.1 --service-role EMR_DefaultRole \
  --ec2-attributes InstanceProfile=EMR_EC2_DefaultRole \
  --instance-fleets \
    InstanceFleetType=MASTER,TargetSpotCapacity=1,\
InstanceTypeConfigs=['{InstanceType=m5.xlarge,BidPrice=0.5}'] \
    InstanceFleetType=CORE,TargetSpotCapacity=1,\
InstanceTypeConfigs=['{InstanceType=m5.xlarge,BidPrice=0.5}']
```

Example Contoh: Primer On-Demand, inti campuran dengan tipe instans tunggal, subnet EC2 tunggal

```
aws emr create-cluster --release-label emr-5.3.1 --service-role EMR_DefaultRole \
  --ec2-attributes InstanceProfile=EMR_EC2_DefaultRole,SubnetIds=['subnet-ab12345c'] \
  --instance-fleets \
    InstanceFleetType=MASTER,TargetOnDemandCapacity=1,\
InstanceTypeConfigs=['{InstanceType=m5.xlarge}'] \
```

```
    InstanceFleetType=CORE,TargetOnDemandCapacity=2,TargetSpotCapacity=6,\
InstanceTypeConfigs=['{InstanceType=m5.xlarge,BidPrice=0.5,WeightedCapacity=2}']
```

Example Contoh: Primer On-Demand, spot core dengan beberapa Jenis instans tertimbang, Timeout untuk Spot, Rentang Subnet EC2

```
aws emr create-cluster --release-label emr-5.3.1 --service-role EMR_DefaultRole \
  --ec2-attributes InstanceProfile=EMR_EC2_DefaultRole,SubnetIds=['subnet-
ab12345c','subnet-de67890f'] \
  --instance-fleets \
    InstanceFleetType=MASTER,TargetOnDemandCapacity=1,\
InstanceTypeConfigs=['{InstanceType=m5.xlarge}'] \
    InstanceFleetType=CORE,TargetSpotCapacity=11,\
InstanceTypeConfigs=['{InstanceType=m5.xlarge,BidPrice=0.5,WeightedCapacity=3}',\
'{InstanceType=m4.2xlarge,BidPrice=0.9,WeightedCapacity=5}'],\
LaunchSpecifications={SpotSpecification='{TimeoutDurationMinutes=120,TimeoutAction=SWITCH_TO_ON
```

Example Contoh: Primer On-Demand, inti campuran, dan tugas dengan beberapa jenis instans tertimbang, batas waktu untuk Instans Spot inti, rentang subnet EC2

```
aws emr create-cluster --release-label emr-5.3.1 --service-role EMR_DefaultRole \
  --ec2-attributes InstanceProfile=EMR_EC2_DefaultRole,SubnetIds=['subnet-
ab12345c','subnet-de67890f'] \
  --instance-fleets \

 InstanceFleetType=MASTER,TargetOnDemandCapacity=1,InstanceTypeConfigs=['{InstanceType=m5.xlarg
 \
    InstanceFleetType=CORE,TargetOnDemandCapacity=8,TargetSpotCapacity=6,\
InstanceTypeConfigs=['{InstanceType=m5.xlarge,BidPrice=0.5,WeightedCapacity=3}',\
'{InstanceType=m4.2xlarge,BidPrice=0.9,WeightedCapacity=5}'],\
LaunchSpecifications={SpotSpecification='{TimeoutDurationMinutes=120,TimeoutAction=SWITCH_TO_ON
 \
    InstanceFleetType=TASK,TargetOnDemandCapacity=3,TargetSpotCapacity=3,\
InstanceTypeConfigs=['{InstanceType=m5.xlarge,BidPrice=0.5,WeightedCapacity=3}']
```

Example Contoh: Spot primer, tidak ada inti atau tugas, konfigurasi Amazon EBS, VPC default

```
aws emr create-cluster --release-label Amazon EMR 5.3.1 --service-role EMR_DefaultRole
 \
  --ec2-attributes InstanceProfile=EMR_EC2_DefaultRole \
  --instance-fleets \
    InstanceFleetType=MASTER,TargetSpotCapacity=1,\
```

```
LaunchSpecifications={SpotSpecification='{TimeoutDurationMinutes=60,TimeoutAction=TERMINATE_CLU
\
InstanceTypeConfigs=['{InstanceType=m5.xlarge,BidPrice=0.5,\
EbsConfiguration={EbsOptimized=true,EbsBlockDeviceConfigs=[{VolumeSpecification={VolumeType=gp2
\
SizeIn GB=100}},{VolumeSpecification={VolumeType=io1,SizeInGB=100,Iop
 s=100},VolumesPerInstance=4}]}}']
```

Example Contoh: Beberapa AMI kustom, beberapa jenis instans, primer sesuai permintaan, inti sesuai permintaan

```
aws emr create-cluster --release-label Amazon EMR 5.3.1 --service-role EMR_DefaultRole
 \
  --ec2-attributes InstanceProfile=EMR_EC2_DefaultRole \
  --instance-fleets \
    InstanceFleetType=MASTER,TargetOnDemandCapacity=1,\
InstanceTypeConfigs=['{InstanceType=m5.xlarge,CustomAmiId=ami-123456},
{InstanceType=m6g.xlarge, CustomAmiId=ami-234567}'] \
    InstanceFleetType=CORE,TargetOnDemandCapacity=1,\
InstanceTypeConfigs=['{InstanceType=m5.xlarge,CustomAmiId=ami-123456},
{InstanceType=m6g.xlarge, CustomAmiId=ami-234567}']
```

Example Contoh: Tambahkan node tugas ke cluster yang sedang berjalan dengan beberapa jenis instance dan beberapa AMI kustom

```
aws emr add-instance-fleet --cluster-id j-123456 --release-label Amazon EMR 5.3.1 \
  --service-role EMR_DefaultRole \
  --ec2-attributes InstanceProfile=EMR_EC2_DefaultRole \
  --instance-fleet \
    InstanceFleetType=Task,TargetSpotCapacity=1,\
InstanceTypeConfigs=['{InstanceType=m5.xlarge,CustomAmiId=ami-123456}',\
'{InstanceType=m6g.xlarge,CustomAmiId=ami-234567}']
```

Example Contoh: Gunakan file konfigurasi JSON

Anda dapat mengkonfigurasi parameter armada instans dalam file JSON, lalu mereferensikan file JSON sebagai parameter tunggal untuk armada instans. Misalnya, perintah berikut merujuk file konfigurasi JSON, *my-fleet-config.json*:

```
aws emr create-cluster --release-label emr-5.30.0 --service-role EMR_DefaultRole \
--ec2-attributes InstanceProfile=EMR_EC2_DefaultRole \
```

```
--instance-fleets file://my-fleet-config.json
```

my-fleet-configFile.json menentukan armada primer, inti, dan instance tugas seperti yang ditunjukkan pada contoh berikut. Armada instance inti menggunakan harga Spot maksimum (`BidPrice`) sebagai persentase On-Demand, sedangkan armada tugas dan instance utama menggunakan harga Spot maksimum (BidPriceAsPercentageofOnDemandPrice) sebagai string dalam USD.

```
[
    {
        "Name": "Masterfleet",
        "InstanceFleetType": "MASTER",
        "TargetSpotCapacity": 1,
        "LaunchSpecifications": {
            "SpotSpecification": {
                "TimeoutDurationMinutes": 120,
                "TimeoutAction": "SWITCH_TO_ON_DEMAND"
            }
        },
        "InstanceTypeConfigs": [
            {
                "InstanceType": "m5.xlarge",
                "BidPrice": "0.89"
            }
        ]
    },
    {
        "Name": "Corefleet",
        "InstanceFleetType": "CORE",
        "TargetSpotCapacity": 1,
        "TargetOnDemandCapacity": 1,
        "LaunchSpecifications": {
          "OnDemandSpecification": {
            "AllocationStrategy": "lowest-price",
            "CapacityReservationOptions":
            {
                "UsageStrategy": "use-capacity-reservations-first",
                "CapacityReservationResourceGroupArn": "String"
            }
        },
            "SpotSpecification": {
                "AllocationStrategy": "capacity-optimized",
                "TimeoutDurationMinutes": 120,
```

```
                "TimeoutAction": "TERMINATE_CLUSTER"
            }
        },
        "InstanceTypeConfigs": [
            {
                "InstanceType": "m5.xlarge",
                "BidPriceAsPercentageOfOnDemandPrice": 100
            }
        ]
    },
    {
        "Name": "Taskfleet",
        "InstanceFleetType": "TASK",
        "TargetSpotCapacity": 1,
        "LaunchSpecifications": {
          "OnDemandSpecification": {
            "AllocationStrategy": "lowest-price",
            "CapacityReservationOptions":
            {
                "CapacityReservationPreference": "none"
            }
        },
            "SpotSpecification": {
                "TimeoutDurationMinutes": 120,
                "TimeoutAction": "TERMINATE_CLUSTER"
            }
        },
        "InstanceTypeConfigs": [
            {
                "InstanceType": "m5.xlarge",
                "BidPrice": "0.89"
            }
        ]
    }
]
```

Ubah kapasitas target untuk armada instans

Gunakan perintah `modify-instance-fleet` untuk menentukan kapasitas target baru armada instans. Anda harus menentukan ID klaster dan ID armada instans. Gunakan perintah `list-instance-fleets` untuk mengambil ID armada instans.

```
aws emr modify-instance-fleet --cluster-id <cluster-id> \
```

```
  --instance-fleet \
    InstanceFleetId='<instance-fleet-id>',TargetOnDemandCapacity=1,TargetSpotCapacity=1
```

Tambahkan armada instans tugas ke klaster

Jika cluster hanya memiliki armada instance primer dan inti, Anda dapat menggunakan `add-instance-fleet` perintah untuk menambahkan armada instance tugas. Anda hanya dapat menggunakan ini untuk menambahkan armada instans tugas.

```
aws emr add-instance-fleet --cluster-id <cluster-id>
  --instance-fleet \
    InstanceFleetType=TASK,TargetSpotCapacity=1,\
LaunchSpecifications={SpotSpecification='{TimeoutDurationMinutes=20,TimeoutAction=TERMINATE_CLU
\
InstanceTypeConfigs=['{InstanceType=m5.xlarge,BidPrice=0.5}']
```

Dapatkan detail konfigurasi armada instans dalam sebuah klaster

Gunakan perintah `list-instance-fleets` untuk mendapatkan detail konfigurasi armada instans dalam sebuah klaster. Perintah mengambil ID klaster sebagai input. Contoh berikut menunjukkan perintah dan outputnya untuk cluster yang berisi grup instance tugas utama dan grup instance tugas inti. Untuk sintaks respons lengkap, lihat ListInstanceFleetsdi Referensi API EMR Amazon.

```
list-instance-fleets --cluster-id <cluster-id>
```

```
{
    "InstanceFleets": [
        {
            "Status": {
                "Timeline": {
                    "ReadyDateTime": 1488759094.637,
                    "CreationDateTime": 1488758719.817
                },
                "State": "RUNNING",
                "StateChangeReason": {
                    "Message": ""
                }
            },
            "ProvisionedSpotCapacity": 6,
            "Name": "CORE",
            "InstanceFleetType": "CORE",
```

```
            "LaunchSpecifications": {
                "SpotSpecification": {
                    "TimeoutDurationMinutes": 60,
                    "TimeoutAction": "TERMINATE_CLUSTER"
                }
            },
            "ProvisionedOnDemandCapacity": 2,
            "InstanceTypeSpecifications": [
                {
                    "BidPrice": "0.5",
                    "InstanceType": "m5.xlarge",
                    "WeightedCapacity": 2
                }
            ],
            "Id": "if-1ABC2DEFGHIJ3"
        },
        {
            "Status": {
                "Timeline": {
                    "ReadyDateTime": 1488759058.598,
                    "CreationDateTime": 1488758719.811
                },
                "State": "RUNNING",
                "StateChangeReason": {
                    "Message": ""
                }
            },
            "ProvisionedSpotCapacity": 0,
            "Name": "MASTER",
            "InstanceFleetType": "MASTER",
            "ProvisionedOnDemandCapacity": 1,
            "InstanceTypeSpecifications": [
                {
                    "BidPriceAsPercentageOfOnDemandPrice": 100.0,
                    "InstanceType": "m5.xlarge",
                    "WeightedCapacity": 1
                }
            ],
           "Id": "if-2ABC4DEFGHIJ4"
        }
    ]
}
```

Gunakan pencadangan kapasitas dengan armada instans

Untuk meluncurkan armada Instans Sesuai Permintaan dengan opsi pencadangan kapasitas, lampirkan izin peran layanan tambahan yang diperlukan guna menggunakan opsi pencadangan kapasitas. Karena opsi pencadangan kapasitas harus digunakan bersama dengan strategi alokasi Sesuai Permintaan, Anda juga harus menyertakan izin yang diperlukan untuk strategi alokasi dalam peran layanan dan kebijakan terkelola Anda. Untuk informasi selengkapnya, lihat Izin strategi alokasi.

Amazon EMR mendukung pencadangan kapasitas terbuka dan tertarget. Topik berikut menunjukkan konfigurasi armada instans yang dapat Anda gunakan dengan tindakan `RunJobFlow` atau perintah `create-cluster` untuk meluncurkan armada instans menggunakan Pencadangan Kapasitas Sesuai Permintaan.

Gunakan pencadangan kapasitas terbuka berdasarkan upaya terbaik

Jika Instans Sesuai Permintaan klaster cocok dengan atribut pencadangan kapasitas terbuka (tipe instans, platform, penghunian, dan Availability Zone) yang tersedia di akun Anda, pencadangan kapasitas akan diterapkan secara otomatis. Namun, tidak ada jaminan bahwa pencadangan kapasitas Anda akan digunakan. Untuk penyediaan klaster, Amazon EMR mengevaluasi semua kumpulan instans yang ditentukan dalam permintaan peluncuran dan menggunakan salah satu dengan harga terendah yang memiliki kapasitas yang memadai untuk meluncurkan semua simpul inti yang diminta. Pencadangan kapasitas terbuka yang tersedia yang cocok dengan kumpulan instans diterapkan secara otomatis. Jika pencadangan kapasitas terbuka yang tersedia tidak cocok dengan kumpulan instans, pencadangan tersebut tetap tidak digunakan.

Setelah simpul inti disediakan, Availability Zone dipilih dan ditetapkan. Amazon EMR menyediakan simpul tugas ke dalam kumpulan instans, dimulai dengan harga yang paling rendah terlebih dahulu, di Availability Zone yang dipilih hingga semua simpul tugas disediakan. Pencadangan kapasitas terbuka yang tersedia yang cocok dengan kumpulan instans diterapkan secara otomatis.

Berikut ini adalah kasus penggunaan logika alokasi kapasitas Amazon EMR untuk menggunakan pencadangan kapasitas terbuka berdasarkan upaya terbaik.

Contoh 1: Kumpulan instans dengan harga terendah dalam permintaan peluncuran memiliki reservasi kapasitas terbuka yang tersedia

Dalam hal ini, Amazon EMR meluncurkan kapasitas di kumpulan instans harga terendah dengan Instans Sesuai Permintaan. Pencadangan kapasitas terbuka Anda yang tersedia di kumpulan instans tersebut digunakan secara otomatis.

On-Demand Strategy	lowest-price		
Requested Capacity	100		
Instance Type	c5.xlarge	m5.xlarge	r5.xlarge
Available Open capacity reservations	150	100	100
On-Demand Price	$	$$	$$$
Instans yang Disediakan	100	-	-
Pencadangan kapasitas terbuka yang digunakan	100	-	-
Pencadangan kapasitas terbuka yang tersedia	50	100	100

Setelah armada instans diluncurkan, Anda dapat menjalankan `describe-capacity-reservations` untuk melihat berapa banyak pencadangan kapasitas tidak terpakai yang tersisa.

Contoh 2: Pool instans dengan harga terendah dalam permintaan peluncuran tidak memiliki reservasi kapasitas terbuka yang tersedia

Dalam hal ini, Amazon EMR meluncurkan kapasitas di kumpulan instans harga terendah dengan Instans Sesuai Permintaan. Namun, pencadangan kapasitas terbuka Anda tetap tidak digunakan.

On-Demand Strategy	lowest-price
Requested Capacity	100

Instance Type	c5.xlarge	m5.xlarge	r5.xlarge
Pencadangan kapasitas terbuka yang tersedia	-	-	100
On-Demand Price	$	$$	$$$
Instans yang Disediakan	100	-	-
Pencadangan kapasitas terbuka yang digunakan	-	-	-
Pencadangan kapasitas terbuka yang tersedia	-	-	100

Konfigurasikan Armada Instance untuk menggunakan reservasi kapasitas terbuka dengan upaya terbaik

Jika Anda menggunakan tindakan `RunJobFlow` untuk membuat klaster berbasis armada instans, atur strategi alokasi Sesuai Permintaan ke `lowest-price` dan `CapacityReservationPreference` untuk opsi pencadangan kapasitas ke `open`.. Atau, jika Anda membiarkan bidang ini kosong, Amazon EMR akan me-default preferensi reservasi kapasitas Instans Sesuai Permintaan. `open`

```
"LaunchSpecifications":
    {"OnDemandSpecification": {
        "AllocationStrategy": "lowest-price",
        "CapacityReservationOptions":
         {
            "CapacityReservationPreference": "open"
         }
        }
    }
```

Anda juga dapat menggunakan CLI Amazon EMR untuk membuat klaster berbasis armada instans menggunakan pencadangan kapasitas terbuka.

```
aws emr create-cluster \
 --name 'open-ODCR-cluster' \
 --release-label emr-5.30.0 \
 --service-role EMR_DefaultRole \
 --ec2-attributes SubnetId=subnet-22XXXX01,InstanceProfile=EMR_EC2_DefaultRole \
 --instance-fleets
 InstanceFleetType=MASTER,TargetOnDemandCapacity=1,InstanceTypeConfigs=['{InstanceType=c4.xlarg
 \

 InstanceFleetType=CORE,TargetOnDemandCapacity=100,InstanceTypeConfigs=['{InstanceType=c5.xlarg
{InstanceType=m5.xlarge},{InstanceType=r5.xlarge}'],\
   LaunchSpecifications={OnDemandSpecification='{AllocationStrategy=lowest-
price,CapacityReservationOptions={CapacityReservationPreference=open}}'}
```

Jika,

- `open-ODCR-cluster` diganti dengan nama klaster menggunakan pencadangan kapasitas terbuka.
- `subnet-22XXXX01` diganti dengan ID subnet.

Gunakan pencadangan kapasitas terbuka terlebih dahulu

Anda dapat memilih untuk mengganti strategi alokasi harga terendah dan memprioritaskan penggunaan pencadangan kapasitas terbuka yang tersedia terlebih dahulu selagi menyediakan klaster Amazon EMR. Dalam hal ini, Amazon EMR mengevaluasi semua kumpulan instans dengan pencadangan kapasitas yang ditentukan dalam permintaan peluncuran dan menggunakan salah satu dengan harga terendah yang memiliki kapasitas memadai untuk meluncurkan semua simpul inti yang diminta. Jika tidak ada kumpulan instans dengan pencadangan kapasitas yang memiliki kapasitas yang memadai untuk simpul inti yang diminta, Amazon EMR kembali ke kasus upaya terbaik yang dijelaskan dalam topik sebelumnya. Artinya, Amazon EMR mengevaluasi ulang semua kumpulan instans yang ditentukan dalam permintaan peluncuran dan menggunakan salah satu dengan harga terendah yang memiliki kapasitas memadai untuk meluncurkan semua simpul inti yang diminta. Pencadangan kapasitas terbuka yang tersedia yang cocok dengan kumpulan instans diterapkan secara otomatis. Jika pencadangan kapasitas terbuka yang tersedia tidak cocok dengan kumpulan instans, pencadangan tersebut tetap tidak digunakan.

Setelah simpul inti disediakan, Availability Zone dipilih dan ditetapkan. Amazon EMR menyediakan simpul tugas ke dalam kumpulan instans dengan pencadangan kapasitas, dimulai dengan yang memiliki harga terendah terlebih dahulu, di Availability Zone yang dipilih hingga semua simpul tugas disediakan. Amazon EMR menggunakan pencadangan kapasitas terbuka yang tersedia yang terdapat di setiap kumpulan instans di Availability Zone yang dipilih terlebih dahulu, dan hanya jika diperlukan, menggunakan strategi harga terendah untuk menyediakan simpul tugas yang lainnya.

Berikut ini adalah kasus penggunaan logika alokasi kapasitas Amazon EMR untuk menggunakan pencadangan kapasitas terbuka terlebih dahulu.

Contoh 1: Kumpulan instans dengan reservasi kapasitas terbuka yang tersedia dalam permintaan peluncuran memiliki kapasitas yang cukup untuk node inti

Dalam hal ini, Amazon EMR meluncurkan kapasitas di kumpulan instans dengan pencadangan kapasitas terbuka yang tersedia terlepas dari harga kumpulan instans. Sehingga, pencadangan kapasitas terbuka Anda digunakan bila memungkinkan, hingga semua simpul inti tersedia.

On-Demand Strategy	lowest-price		
Requested Capacity	100		
Usage Strategy	use-capacity-reservations-first		
Instance Type	c5.xlarge	m5.xlarge	r5.xlarge
Available Open capacity reservations	-	-	150
On-Demand Price	$	$$	$$$
Instans yang Disediakan	-	-	100
Pencadangan kapasitas terbuka yang digunakan	-	-	100
Pencadangan kapasitas terbuka yang tersedia	-	-	50

Contoh 2: Kumpulan instans dengan reservasi kapasitas terbuka yang tersedia dalam permintaan peluncuran tidak memiliki kapasitas yang cukup untuk node inti

Dalam hal ini, Amazon EMR kembali meluncurkan simpul inti menggunakan strategi harga terendah dengan upaya terbaik untuk menggunakan pencadangan kapasitas.

On-Demand Strategy	lowest-price		
Requested Capacity	100		
Usage Strategy	use-capacity-reservations-first		
Instance Type	c5.xlarge	m5.xlarge	r5.xlarge
Available Open capacity reservations	10	50	50
On-Demand Price	$	$$	$$$
Instans yang Disediakan	100	-	-
Pencadangan kapasitas terbuka yang digunakan	10	-	-
Pencadangan kapasitas terbuka yang tersedia	-	50	50

Setelah armada instans diluncurkan, Anda dapat menjalankan `describe-capacity-reservations` untuk melihat berapa banyak pencadangan kapasitas tidak terpakai yang tersisa.

Konfigurasikan Armada Instance untuk menggunakan reservasi kapasitas terbuka terlebih dahulu

Jika Anda menggunakan tindakan `RunJobFlow` untuk membuat klaster berbasis armada instans, atur strategi alokasi Sesuai Permintaan ke `lowest-price` dan `UsageStrategy` untuk `CapacityReservationOptions` hingga `use-capacity-reservations-first`.

```
"LaunchSpecifications":
```

```
    {"OnDemandSpecification": {
        "AllocationStrategy": "lowest-price",
        "CapacityReservationOptions":
         {
            "UsageStrategy": "use-capacity-reservations-first"
         }
       }
    }
```

Anda juga dapat menggunakan Amazon EMR CLI untuk membuat cluster berbasis armada instans menggunakan reservasi kapasitas terlebih dahulu.

```
aws emr create-cluster \
  --name 'use-CR-first-cluster' \
  --release-label emr-5.30.0 \
  --service-role EMR_DefaultRole \
  --ec2-attributes SubnetId=subnet-22XXXX01,InstanceProfile=EMR_EC2_DefaultRole \
  --instance-fleets \

 InstanceFleetType=MASTER,TargetOnDemandCapacity=1,InstanceTypeConfigs=['{InstanceType=c4.xlarg
 \

 InstanceFleetType=CORE,TargetOnDemandCapacity=100,InstanceTypeConfigs=['{InstanceType=c5.xlarg
{InstanceType=m5.xlarge},{InstanceType=r5.xlarge}'],\
LaunchSpecifications={OnDemandSpecification='{AllocationStrategy=lowest-
price,CapacityReservationOptions={UsageStrategy=use-capacity-reservations-first}}'}
```

Jika,

- `use-CR-first-cluster` diganti dengan nama klaster menggunakan pencadangan kapasitas terbuka.
- `subnet-22XXXX01` diganti dengan ID subnet.

Gunakan pencadangan kapasitas yang ditargetkan terlebih dahulu

Saat Anda menyediakan klaster EMR Amazon, Anda dapat memilih untuk mengganti strategi alokasi harga terendah dan memprioritaskan menggunakan reservasi kapasitas bertarget yang tersedia terlebih dahulu. Dalam hal ini, Amazon EMR mengevaluasi semua kumpulan instans dengan pencadangan kapasitas yang ditargetkan yang ditentukan dalam permintaan peluncuran dan memilih satu dengan harga terendah yang memiliki kapasitas memadai untuk meluncurkan

semua simpul inti yang diminta. Jika tidak ada kumpulan instans dengan pencadangan kapasitas yang ditargetkan memiliki kapasitas yang memadai untuk simpul inti, Amazon EMR kembali ke kasus upaya terbaik yang dijelaskan sebelumnya. Artinya, Amazon EMR mengevaluasi ulang semua kumpulan instans yang ditentukan dalam permintaan peluncuran dan memilih satu dengan harga terendah yang memiliki kapasitas memadai untuk meluncurkan semua simpul inti yang diminta. Pencadangan kapasitas terbuka tersedia yang cocok dengan kumpulan instans diterapkan secara otomatis. Namun, pencadangan kapasitas yang ditargetkan tetap tidak digunakan.

Setelah simpul inti disediakan, Availability Zone dipilih dan ditetapkan. Amazon EMR menyediakan simpul tugas ke dalam kumpulan instans dengan pencadangan kapasitas yang ditargetkan, dimulai dengan yang memiliki harga terenda terlebih dahulu, di Availability Zone yang dipilih hingga semua simpul tugas disediakan. Amazon EMR mencoba menggunakan pencadangan kapasitas tertarget yang tersedia yang berada di setiap kumpulan instans di Availability Zone yang dipilih terlebih dahulu. Kemudian, hanya jika diperlukan, Amazon EMR menggunakan strategi harga terendah untuk menyediakan simpul tugas lainnya.

Berikut ini adalah kasus penggunaan logika alokasi kapasitas Amazon EMR untuk menggunakan pencadangan kapasitas yang ditargetkan terlebih dahulu.

Contoh 1: Kumpulan instans dengan reservasi kapasitas tertarget yang tersedia dalam permintaan peluncuran memiliki kapasitas yang cukup untuk node inti

Dalam hal ini, Amazon EMR meluncurkan kapasitas di kumpulan instans dengan pencadangan kapasitas tertarget yang tersedia terlepas dari harga kumpulan instans. Sehingga, pencadangan kapasitas yang Anda targetkan digunakan bila memungkinkan hingga semua simpul inti tersedia.

On-Demand Strategy	lowest-price		
Usage Strategy	use-capacity-reservations-first		
Requested Capacity	100		
Instance Type	c5.xlarge	m5.xlarge	r5.xlarge
Available targeted capacity reservations	-	-	150
On-Demand Price	$	$$	$$$

Instans yang Disediakan	-	-	100
Pencadangan kapasitas tertarget yang digunakan	-	-	100
Pencadangan kapasitas tertarget yang tersedia	-	-	50

Example Contoh 2: Kumpulan instans dengan pencadangan kapasitas tertarget yang tersedia dalam permintaan peluncuran tidak memiliki kapasitas yang memadai untuk simpul inti

On-Demand Strategy	lowest-price		
Requested Capacity	100		
Usage Strategy	use-capacity-reservations-first		
Instance Type	c5.xlarge	m5.xlarge	r5.xlarge
Available targeted capacity reservations	10	50	50
On-Demand Price	$	$$	$$$
Instans yang Disediakan	100	-	-
Reservasi kapasitas yang ditargetkan digunakan	10	-	-
Pencadangan kapasitas tertarget yang tersedia	-	50	50

Setelah armada instans diluncurkan, Anda dapat menjalankan `describe-capacity-reservations` untuk melihat berapa banyak pencadangan kapasitas tidak terpakai yang tersisa.

Konfigurasikan Armada Instance untuk menggunakan reservasi kapasitas yang ditargetkan terlebih dahulu

Jika Anda menggunakan tindakan `RunJobFlow` untuk membuat klaster berbasis armada instans, atur strategi alokasi Sesuai Permintaan ke `lowest-price`, `UsageStrategy` untuk `CapacityReservationOptions` hingga `use-capacity-reservations-first`, dan `CapacityReservationResourceGroupArn` hingga `CapacityReservationOptions` ke `<your resource group ARN>`. Untuk informasi selengkapnya, lihat Bekerja dengan pencadangan kapasitas di Panduan Pengguna Amazon EC2 untuk Instans Linux.

```
"LaunchSpecifications":
    {"OnDemandSpecification": {
        "AllocationStrategy": "lowest-price",
        "CapacityReservationOptions":
         {
            "UsageStrategy": "use-capacity-reservations-first",
            "CapacityReservationResourceGroupArn": "arn:aws:resource-groups:sa-
east-1:123456789012:group/MyCRGroup"
         }
       }
    }
```

Jika `arn:aws:resource-groups:sa-east-1:123456789012:group/MyCRGroup` diganti dengan grup sumber daya Anda ARN.

Anda juga dapat menggunakan Amazon EMR CLI untuk membuat cluster berbasis armada instance menggunakan reservasi kapasitas yang ditargetkan.

```
aws emr create-cluster \
  --name 'targeted-CR-cluster' \
  --release-label emr-5.30.0 \
  --service-role EMR_DefaultRole \
  --ec2-attributes SubnetId=subnet-22XXXX01,InstanceProfile=EMR_EC2_DefaultRole \
  --instance-fleets
 InstanceFleetType=MASTER,TargetOnDemandCapacity=1,InstanceTypeConfigs=['{InstanceType=c4.xlarg
 \
    InstanceFleetType=CORE,TargetOnDemandCapacity=100,\
InstanceTypeConfigs=['{InstanceType=c5.xlarge},{InstanceType=m5.xlarge},
{InstanceType=r5.xlarge}'],\
```

```
LaunchSpecifications={OnDemandSpecification='{AllocationStrategy=lowest-
price,CapacityReservationOptions={UsageStrategy=use-capacity-reservations-
first,CapacityReservationResourceGroupArn=arn:aws:resource-groups:sa-
east-1:123456789012:group/MyCRGroup}}'}
```

Di mana,

- `targeted-CR-cluster` diganti dengan nama klaster Anda menggunakan pencadangan kapasitas yang ditargetkan.
- `subnet-22XXXX01` diganti dengan ID subnet.
- `arn:aws:resource-groups:sa-east-1:123456789012:group/MyCRGroup` diganti dengan grup sumber daya ARN.

Hindari menggunakan pencadangan kapasitas terbuka yang tersedia

Example

Jika Anda ingin menghindari penggunaan pencadangan kapasitas terbuka secara tidak terduga saat meluncurkan klaster Amazon EMR, atur strategi alokasi Sesuai Permintaan ke `lowest-price` dan `CapacityReservationPreference` untuk `CapacityReservationOptions` hingga `none`. Jika tidak, Amazon EMR menetapkan preferensi pencadangan kapasitas Instans Sesuai Permintaan ke default `open` dan mencoba menggunakan pencadangan kapasitas terbuka yang tersedia berdasarkan upaya terbaik.

```
"LaunchSpecifications":
    {"OnDemandSpecification": {
        "AllocationStrategy": "lowest-price",
        "CapacityReservationOptions":
         {
            "CapacityReservationPreference": "none"
         }
       }
    }
```

Anda juga dapat menggunakan CLI Amazon EMR untuk membuat klaster berbasis armada instans tanpa menggunakan pencadangan kapasitas terbuka apa pun.

```
aws emr create-cluster \
  --name 'none-CR-cluster' \
```

```
  --release-label emr-5.30.0 \
  --service-role EMR_DefaultRole \
  --ec2-attributes SubnetId=subnet-22XXXX01,InstanceProfile=EMR_EC2_DefaultRole \
  --instance-fleets \

 InstanceFleetType=MASTER,TargetOnDemandCapacity=1,InstanceTypeConfigs=['{InstanceType=c4.xlarg
 \

 InstanceFleetType=CORE,TargetOnDemandCapacity=100,InstanceTypeConfigs=['{InstanceType=c5.xlarg
{InstanceType=m5.xlarge},{InstanceType=r5.xlarge}'],\
LaunchSpecifications={OnDemandSpecification='{AllocationStrategy=lowest-
price,CapacityReservationOptions={CapacityReservationPreference=none}}'}
```

Jika,

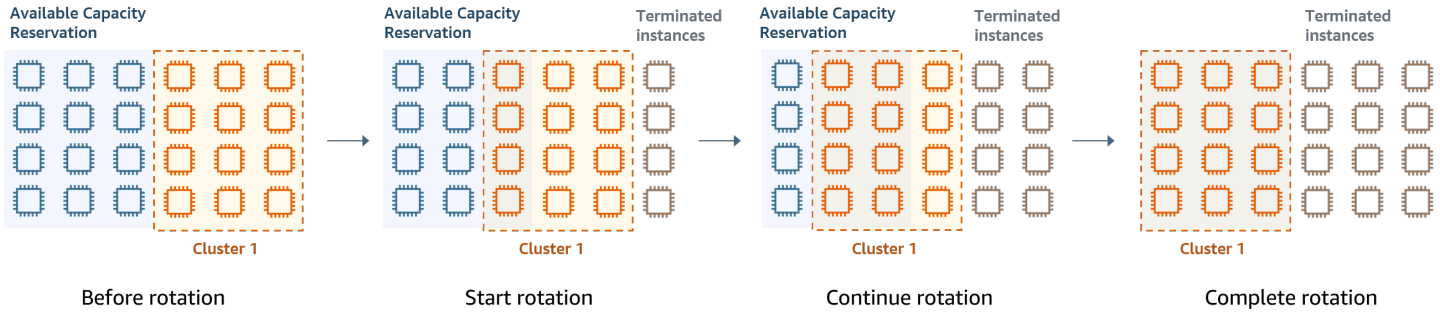
- `none-CR-cluster` diganti dengan nama klaster Anda yang tidak menggunakan pencadangan kapasitas terbuka.
- `subnet-22XXXX01` diganti dengan ID subnet.

Skenario untuk menggunakan pencadangan kapasitas

Anda bisa mendapatkan keuntungan dari penggunaan pencadangan kapasitas dalam skenario berikut.

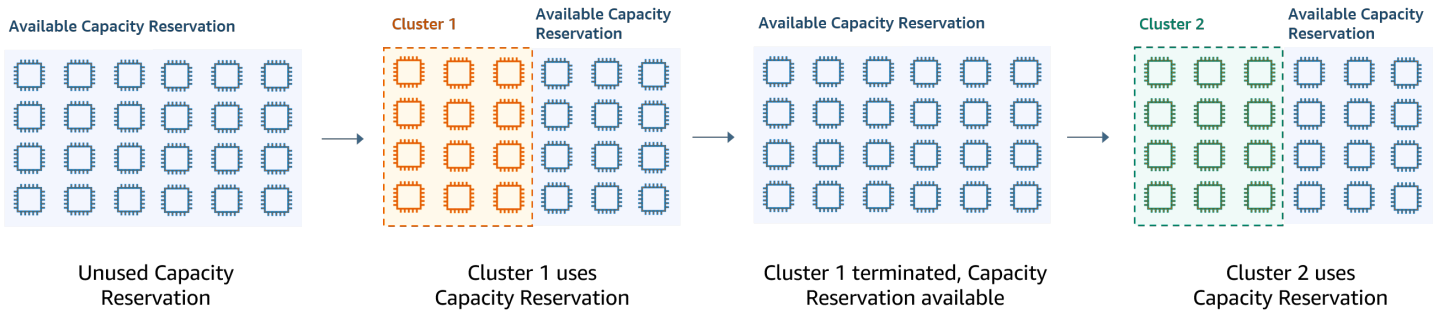
Skenario 1: Rotasi klaster yang berjalan lama menggunakan pencadangan kapasitas

Saat merotasi klaster yang berjalan lama, Anda mungkin memiliki persyaratan ketat mengenai tipe instans dan Availability Zone untuk instans baru yang Anda sediakan. Dengan pencadangan kapasitas, Anda dapat menggunakan jaminan kapasitas untuk menyelesaikan rotasi klaster tanpa gangguan.



Skenario 2: Sediakan klaster jangka pendek berturut-turut menggunakan pencadangan kapasitas

Anda juga dapat menggunakan pencadangan kapasitas untuk menyediakan sekelompok klaster jangka pendek yang berurutan untuk beban kerja individual sehingga saat Anda mengakhiri klaster, klaster berikutnya dapat menggunakan pencadangan kapasitas. Anda dapat menggunakan pencadangan kapasitas yang ditargetkan untuk memastikan bahwa hanya klaster yang dituju yang menggunakan pencadangan kapasitas.



Konfigurasi grup instans seragam

Dengan konfigurasi grup instans, setiap jenis simpul (utama, inti, atau tugas) terdiri dari tipe instans yang sama dan opsi pembelian yang sama untuk instans: Sesuai Permintaan atau Spot. Anda menentukan setelan ini saat membuat grup instans. Mereka tidak bisa diubah nanti. Namun, Anda dapat menambahkan instans dengan jenis dan opsi pembelian yang sama ke grup instans inti dan tugas. Anda juga dapat menghapus instans.

Jika Instans Sesuai Permintaan klaster cocok dengan atribut pencadangan kapasitas terbuka (tipe instans, platform, penghunian, dan Availability Zone) yang tersedia di akun Anda, pencadangan kapasitas akan diterapkan secara otomatis. Anda dapat menggunakan reservasi kapasitas terbuka untuk node primer, inti, dan tugas. Namun, Anda tidak dapat menggunakan pencadangan kapasitas yang ditargetkan atau mencegah instans diluncurkan ke pencadangan kapasitas terbuka dengan atribut yang cocok saat Anda menyediakan klaster menggunakan grup instans. Jika Anda ingin menggunakan pencadangan kapasitas yang ditargetkan atau mencegah instans diluncurkan ke pencadangan kapasitas terbuka, gunakan Armada Instans. Untuk informasi selengkapnya, lihat Gunakan pencadangan kapasitas dengan armada instans.

Untuk menambahkan tipe instans yang berbeda setelah klaster dibuat, Anda dapat menambahkan grup instans tugas tambahan. Anda dapat memilih tipe instans dan opsi pembelian yang berbeda untuk setiap grup instans. Untuk informasi selengkapnya, lihat Gunakan penskalaan cluster.

Saat meluncurkan instans, preferensi pencadangan kapasitas Instans Berdasarkan Permintaan akan default ke `open`, yang memungkinkannya dijalankan di pencadangan kapasitas terbuka yang memiliki atribut yang cocok (tipe instans, platform, Availability Zone). Untuk informasi lebih lanjut

tentang Pencadangan Kapasitas Sesuai Permintaan, lihat Gunakan pencadangan kapasitas dengan armada instans.

Bagian ini mencakup pembuatan klaster dengan grup instans seragam. Untuk informasi lebih lanjut tentang memodifikasi grup instans yang ada dengan menambahkan atau menghapus instans secara manual atau dengan penskalaan otomatis, lihat Mengelola klaster.

Gunakan konsol untuk mengkonfigurasi grup instans seragam

> **Note**
>
> Kami telah mendesain ulang konsol EMR Amazon agar lebih mudah digunakan. Lihat Apa yang baru dengan konsol? untuk mempelajari tentang perbedaan antara pengalaman konsol lama dan baru.

New console

Untuk membuat cluster dengan grup instance dengan konsol baru

1. Masuk keAWS Management Console, dan buka konsol EMR Amazon di https://console.aws.amazon.com/emr.
2. Di bawah EMR pada EC2 di panel navigasi kiri, pilih Clusters, dan pilih Create cluster.
3. Di bawah konfigurasi Cluster, pilih Grup instans.
4. Di bawah grup Node, ada bagian untuk setiap jenis grup node. Untuk grup simpul primer, pilih kotak centang Gunakan beberapa node primer jika Anda ingin memiliki 3 node primer. Pilih kotak centang opsi Gunakan pembelian Spot jika Anda ingin menggunakan pembelian Spot.
5. Untuk grup node primer dan inti, pilih Add instance type dan pilih hingga 5 tipe instance. Untuk grup tugas, pilih Tambahkan jenis instans dan pilih hingga lima belas jenis instans. Amazon EMR mungkin menyediakan campuran jenis instans ini saat meluncurkan cluster.
6. Di bawah setiap tipe grup node, pilih menu tarik-turun Tindakan di samping setiap instance untuk mengubah pengaturan ini:

   Tambahkan volume EBS

   Tentukan volume EBS yang akan dilampirkan ke jenis instans setelah Amazon EMR menyediakannya.

Edit harga Spot maksimum

Tentukan harga Spot maksimum untuk setiap jenis instans dalam armada. Anda dapat menetapkan harga ini sebagai persentase dari harga Sesuai Permintaan, atau sebagai jumlah dolar tertentu. Jika harga Spot saat ini di Availability Zone di bawah harga Spot maksimum Anda, Amazon EMR menyediakan Instans Spot. Anda tidak selalu membayar harga Spot dengan harga Spot maksimum.

7. Secara opsional, perluas konfigurasi Node untuk memasukkan konfigurasi JSON atau memuat JSON dari Amazon S3.
8. Pilih opsi lain yang berlaku untuk cluster Anda.
9. Untuk meluncurkan klaster Anda, pilih Buat klaster.

Old console

Prosedur berikut meliputi Opsi lanjutan saat Anda membuat klaster. Menggunakan Opsi cepat juga membuat klaster dengan konfigurasi grup instans.

Untuk membuat cluster dengan grup instance seragam dengan konsol lama

1. Arahkan ke konsol EMR Amazon baru dan pilih Beralih ke konsol lama dari navigasi samping. Untuk informasi selengkapnya tentang apa yang diharapkan saat beralih ke konsol lama, lihat Menggunakan konsol lama.
2. Pilih Buat klaster.
3. Pilih Pergi ke opsi lanjutan, masukkan opsi Konfigurasi perangkat lunak, lalu pilih Selanjutnya.
4. Di layar Pengaturan perangkat keras, biarkan Grup instans seragam dipilih.
5. Pilih Jaringan, lalu pilihSubnet EC2 untuk klaster Anda. Subnet yang Anda pilih dikaitkan dengan Grup Ketersediaan, yang dicantumkan pada setiap subnet. Untuk informasi selengkapnya, lihat Mengkonfigurasi jaringan.

> **Note**
>
> Akun dan Wilayah Anda dapat memberi Anda opsi untuk memilih Luncurkan ke EC2-Classic untuk Jaringan. Jika Anda memilih opsi tersebut, pilih Availability Zone EC2 bukan Subnet EC2. Untuk informasi selengkapnya, lihat Amazon EC2 dan Amazon VPC di Panduan Pengguna Amazon EC2 untuk Instans Linux.

6. Dalam setiap baris Jenis simpul:

   - Di bawah Jenis simpul, jika Anda ingin mengubah nama default grup instans, pilih ikon pensil lalu masukkan nama yang familiar. Jika ingin menghapus grup instans Tugas, pilih ikon X. Pilih Tambah grup instance tugas untuk menambahkan grup instans Tugas tambahan.
   - Di bawah Tipe instans pilih ikon pensil lalu pilih tipe instans yang ingin Anda gunakan untuk jenis simpul tersebut.

     > **Important**
     >
     > Saat Anda memilih tipe instans menggunakan AWS Management Console, jumlah vCPU yang ditampilkan untuk setiap Tipe instans adalah jumlah vcore YARN untuk tipe instans tersebut, bukan jumlah vCPU EC2 untuk tipe instans tersebut. Untuk informasi selengkapnya tentang jumlah vCPU untuk setiap tipe instans, lihat Tipe Instans Amazon EC2.

   - Di bawah Jenis instans, pilih ikon pensil untuk Konfigurasi dan kemudian edit konfigurasi untuk aplikasi untuk setiap grup instans.
   - Di bawah Jumlah instans, masukkan jumlah instans yang akan digunakan untuk setiap jenis simpul.
   - Di bawah Opsi pembelian, pilih Sesuai Permintaan atau Spot. Jika Anda memilih Spot, pilih opsi harga maksimum untuk Instans Spot. Secara default, Gunakan Sesuai Permintaan sebagai harga maks dipilih. Anda dapat memilih Set maks $/jam lalu masukkan harga maksimum Anda. Availability Zone Subnet EC2 yang Anda pilih adalah di bawah Harga Spot maksimum.

     > **Tip**
     >
     > Jeda pada tooltip informasi untuk Spot untuk melihat harga Spot saat ini untuk Availability Zone di Wilayah saat ini. Harga Spot terendah berwarna hijau. Anda mungkin ingin menggunakan informasi ini untuk mengubah pilihan Subnet EC2 Anda.

   - Di bawah Auto Scaling untuk jenis simpul Inti dan Tugas, pilih ikon pensil, lalu konfigurasi opsi auto scaling. Untuk informasi selengkapnya, lihat Menggunakan penskalaan otomatis dengan kebijakan kustom untuk grup instans .

7. Pilih Tambah grup instans tugas seperti yang diinginkan lalu konfigurasi pengaturan seperti yang dijelaskan pada langkah sebelumnya.
8. PilihSelanjutnya, ubah pengaturan klaster lainnya, lalu luncurkan klaster.

Gunakan AWS CLI untuk membuat klaster dengan grup instans seragam

Untuk menentukan konfigurasi grup instans klaster dengan menggunakan AWS CLI, gunakan perintah `create-cluster` bersama dengan parameter `--instance-groups`. Amazon EMR mengasumsikan opsi Instans Sesuai Permintaan kecuali Anda menentukan argumen `BidPrice` untuk grup instans. Untuk contoh perintah `create-cluster` yang meluncurkan grup instans seragam dengan Instans Sesuai Permintaan dan berbagai opsi klaster, ketik `aws emr create-cluster help` di baris perintah, atau lihat buat-klaster di AWS CLI Referensi Perintah.

Anda dapat menggunakan AWS CLI untuk membuat grup instans seragam dalam klaster yang menggunakan Instans Spot. Harga Spot yang ditawarkan tergantung pada Availability Zone. Saat Anda menggunakan CLI atau API, Anda dapat menentukan Availability Zone baik dengan argumen `AvailabilityZone` (jika Anda menggunakan jaringan klasik EC2) atau `SubnetID` argumen dari `--ec2-attributes` parameter. Availability Zone atau subnet yang Anda pilih berlaku untuk klaster, sehingga digunakan untuk semua grup instans. Jika Anda tidak menentukan Availability Zone atau subnet secara jelas, Amazon EMR akan memilih Availability Zone dengan harga Spot terendah saat meluncurkan klaster.

Contoh berikut menunjukkan `create-cluster` perintah yang menciptakan primer, inti, dan dua kelompok instance tugas yang semuanya menggunakan Instans Spot. Ganti *`myKey dengan nama key`* pair Amazon EC2 Anda.

> **Note**
>
> Karakter lanjutan baris Linux (\) disertakan agar mudah dibaca Karakter ini bisa dihapus atau digunakan dalam perintah Linux. Untuk Windows, hapus atau ganti dengan tanda sisipan (^).

```
aws emr create-cluster --name "MySpotCluster" \
  --release-label emr-7.0.0 \
  --use-default-roles \
  --ec2-attributes KeyName=myKey \
  --instance-groups \
    InstanceGroupType=MASTER,InstanceType=m5.xlarge,InstanceCount=1,BidPrice=0.25 \
```

```
    InstanceGroupType=CORE,InstanceType=m5.xlarge,InstanceCount=2,BidPrice=0.03 \
    InstanceGroupType=TASK,InstanceType=m5.xlarge,InstanceCount=4,BidPrice=0.03 \
    InstanceGroupType=TASK,InstanceType=m5.xlarge,InstanceCount=2,BidPrice=0.04
```

Dengan menggunakan CLI, Anda dapat membuat cluster grup instance seragam yang menentukan AMI kustom unik untuk setiap jenis instans dalam grup instans. Ini memungkinkan Anda untuk menggunakan arsitektur instance yang berbeda dalam grup instance yang sama. Setiap jenis instans harus menggunakan AMI kustom dengan arsitektur yang cocok. Misalnya, Anda akan mengonfigurasi tipe instans m5.xlarge dengan AMI kustom arsitektur x86_64, dan tipe instans m6g.xlarge dengan AMI kustom arsitektur (ARM) yang sesuai. `AWS  AARCH64`

Contoh berikut menunjukkan cluster grup instance seragam yang dibuat dengan dua tipe instance, masing-masing dengan AMI kustomnya sendiri. Perhatikan bahwa AMI kustom ditentukan hanya pada tingkat tipe instance, bukan pada tingkat cluster. Ini untuk menghindari konflik antara AMI tipe instance dan AMI di tingkat cluster, yang akan menyebabkan peluncuran cluster gagal.

```
aws emr create-cluster
  --release-label emr-5.30.0 \
  --service-role EMR_DefaultRole \
  --ec2-attributes SubnetId=subnet-22XXXX01,InstanceProfile=EMR_EC2_DefaultRole \
  --instance-groups \

 InstanceGroupType=MASTER,InstanceType=m5.xlarge,InstanceCount=1,CustomAmiId=ami-123456
 \

 InstanceGroupType=CORE,InstanceType=m6g.xlarge,InstanceCount=1,CustomAmiId=ami-234567
```

Anda dapat menambahkan beberapa AMI kustom ke grup instans yang Anda tambahkan ke cluster yang sedang berjalan. `CustomAmiId`Argumen dapat digunakan dengan `add-instance-groups` perintah seperti yang ditunjukkan pada contoh berikut.

```
aws emr add-instance-groups --cluster-id j-123456 \
  --instance-groups \

 InstanceGroupType=Task,InstanceType=m5.xlarge,InstanceCount=1,CustomAmiId=ami-123456
```

Gunakan Java SDK untuk membuat grup instans

Anda memulai objek `InstanceGroupConfig` yang menentukan konfigurasi grup instans untuk klaster. Untuk menggunakan Instans Spot, atur properti `withBidPrice` dan `withMarket` pada

objek `InstanceGroupConfig`. Kode berikut menunjukkan cara mendefinisikan grup instance primer, inti, dan tugas yang menjalankan Instans Spot.

```
InstanceGroupConfig instanceGroupConfigMaster = new InstanceGroupConfig()
 .withInstanceCount(1)
 .withInstanceRole("MASTER")
 .withInstanceType("m4.large")
 .withMarket("SPOT")
 .withBidPrice("0.25");

InstanceGroupConfig instanceGroupConfigCore = new InstanceGroupConfig()
 .withInstanceCount(4)
 .withInstanceRole("CORE")
 .withInstanceType("m4.large")
 .withMarket("SPOT")
 .withBidPrice("0.03");

InstanceGroupConfig instanceGroupConfigTask = new InstanceGroupConfig()
 .withInstanceCount(2)
 .withInstanceRole("TASK")
 .withInstanceType("m4.large")
 .withMarket("SPOT")
 .withBidPrice("0.10");
```

Praktik terbaik misalnya dan fleksibilitas Availability Zone

Masing-masing Wilayah AWS memiliki beberapa lokasi terisolasi yang dikenal sebagai Availability Zones. Saat meluncurkan instance, Anda dapat secara opsional menentukan Availability Zone (AZ) di Wilayah AWS yang Anda gunakan. Fleksibilitas Availability Zone adalah distribusi instans di beberapa AZ. Jika satu instance gagal, Anda dapat mendesain aplikasi sehingga instance di AZ lain dapat menangani permintaan. Untuk informasi selengkapnya tentang Availability Zone, lihat dokumentasi Wilayah dan zona di Panduan Pengguna Amazon EC2.

Fleksibilitas instans adalah penggunaan beberapa jenis instance untuk memenuhi persyaratan kapasitas. Bila Anda mengekspresikan fleksibilitas dengan instans, Anda dapat menggunakan kapasitas agregat di seluruh ukuran instans, keluarga, dan generasi. Fleksibilitas yang lebih besar meningkatkan peluang untuk menemukan dan mengalokasikan jumlah kapasitas komputasi yang Anda butuhkan jika dibandingkan dengan cluster yang menggunakan satu jenis instans.

Fleksibilitas Instance dan Availability Zone mengurangi kesalahan kapasitas (ICE) dan interupsi Spot yang tidak mencukupi jika dibandingkan dengan cluster dengan tipe instans tunggal atau

AZ. Gunakan praktik terbaik yang dibahas di sini untuk menentukan contoh mana yang akan didiversifikasi setelah Anda mengetahui keluarga dan ukuran instans awal. Pendekatan ini memaksimalkan ketersediaan ke kumpulan kapasitas Amazon EC2 dengan kinerja minimal dan varians biaya.

Menjadi fleksibel tentang Availability Zone

Kami menyarankan Anda mengonfigurasi semua Availability Zone untuk digunakan di virtual private cloud (VPC) dan Anda memilihnya untuk klaster EMR Anda. Cluster harus ada hanya dalam satu Availability Zone, tetapi dengan armada instans EMR Amazon, Anda dapat memilih beberapa subnet untuk Availability Zone yang berbeda. Saat Amazon EMR meluncurkan cluster, ia melihat subnet tersebut untuk menemukan instance dan opsi pembelian yang Anda tentukan. Saat Anda menyediakan kluster EMR untuk beberapa subnet, kluster Anda dapat mengakses kumpulan kapasitas Amazon EC2 yang lebih dalam jika dibandingkan dengan cluster dalam satu subnet.

Jika Anda harus memprioritaskan sejumlah Availability Zone untuk digunakan di virtual private cloud (VPC) untuk klaster EMR Anda, Anda dapat memanfaatkan kemampuan skor penempatan Spot dengan Amazon EC2. Dengan penilaian penempatan Spot, Anda menentukan persyaratan komputasi untuk Instans Spot Anda, lalu EC2 mengembalikan sepuluh besar Wilayah AWS atau Availability Zone yang dicetak pada skala dari 1 hingga 10. Skor 10 menunjukkan bahwa permintaan Spot Anda sangat mungkin berhasil; skor 1 menunjukkan bahwa permintaan Spot Anda tidak mungkin berhasil. Untuk informasi selengkapnya tentang cara menggunakan penilaian penempatan Spot, lihat Skor penempatan spot di Panduan Pengguna Amazon EC2 untuk Instans Linux.

Menjadi fleksibel tentang jenis instance

Fleksibilitas instans adalah penggunaan beberapa jenis instance untuk memenuhi persyaratan kapasitas. Fleksibilitas instans menguntungkan penggunaan Instans Amazon EC2 Spot dan On-Demand. Dengan Instans Spot, fleksibilitas instans memungkinkan Amazon EC2 meluncurkan instans dari kumpulan kapasitas yang lebih dalam menggunakan data kapasitas waktu nyata. Ini juga memprediksi contoh mana yang paling tersedia. Ini menawarkan lebih sedikit gangguan dan dapat mengurangi biaya keseluruhan beban kerja. Dengan Instans Sesuai Permintaan, fleksibilitas instans mengurangi kesalahan kapasitas (ICE) yang tidak mencukupi saat penyediaan kapasitas total di sejumlah besar kumpulan instans.

Untuk klaster Grup Instance, Anda dapat menentukan hingga 50 jenis instans EC2. Untuk Armada Instance dengan strategi alokasi, Anda dapat menentukan hingga 30 jenis instans EC2 untuk setiap grup node primer, inti, dan tugas. Berbagai contoh yang lebih luas meningkatkan manfaat fleksibilitas instans.

Mengekspresikan fleksibilitas contoh

Pertimbangkan praktik terbaik berikut untuk mengekspresikan fleksibilitas instans untuk aplikasi Anda.

Topik

- Tentukan contoh keluarga dan ukuran
- Sertakan contoh tambahan

Tentukan contoh keluarga dan ukuran

Amazon EMR mendukung beberapa jenis instans untuk kasus penggunaan yang berbeda. Jenis instance ini tercantum dalam Tipe instans yang didukung dokumentasi. Setiap jenis instance milik keluarga instance yang menjelaskan aplikasi apa yang dioptimalkan untuk jenis aplikasi tersebut.

Untuk beban kerja baru, Anda harus melakukan benchmark dengan tipe instance dalam keluarga tujuan umum, seperti `m5` atau. `c5` Kemudian, pantau metrik OS dan YARN dari Ganglia dan Amazon CloudWatch untuk menentukan kemacetan sistem pada beban puncak. Kemacetan termasuk CPU, memori, penyimpanan, dan operasi I/O. Setelah Anda mengidentifikasi kemacetan, pilih komputasi yang dioptimalkan, dioptimalkan memori, penyimpanan dioptimalkan, atau kelompok instans lain yang sesuai untuk jenis instans Anda. Untuk detail selengkapnya, lihat halaman Tentukan infrastruktur yang tepat untuk beban kerja Spark Anda di panduan praktik terbaik Amazon EMR. GitHub

Selanjutnya, identifikasi wadah YARN terkecil atau pelaksana Spark yang dibutuhkan aplikasi Anda. Ini adalah ukuran instance terkecil yang sesuai dengan wadah dan ukuran instance minimum untuk cluster. Gunakan metrik ini untuk menentukan contoh yang dapat Anda diversifikasi lebih lanjut. Contoh yang lebih kecil akan memungkinkan lebih banyak fleksibilitas instance.

Untuk fleksibilitas contoh maksimum, Anda harus memanfaatkan sebanyak mungkin contoh. Kami menyarankan Anda melakukan diversifikasi dengan instance yang memiliki spesifikasi perangkat keras serupa. Ini memaksimalkan akses ke kumpulan kapasitas EC2 dengan biaya minimal dan varians kinerja. Diversifikasi lintas ukuran. Untuk melakukannya, prioritaskan AWS Graviton dan generasi sebelumnya terlebih dahulu. Sebagai aturan umum, cobalah untuk fleksibel di setidaknya 15 jenis instans untuk setiap beban kerja. Kami menyarankan Anda memulai dengan instans tujuan umum, komputasi yang dioptimalkan, atau dioptimalkan memori. Jenis contoh ini akan memberikan fleksibilitas terbesar.

Sertakan contoh tambahan

Untuk keragaman maksimum, sertakan jenis instance tambahan. Prioritaskan ukuran instans, Graviton, dan fleksibilitas generasi terlebih dahulu. Ini memungkinkan akses ke kumpulan kapasitas EC2 tambahan dengan profil biaya dan kinerja yang serupa. Jika Anda membutuhkan fleksibilitas lebih lanjut karena gangguan ICE atau spot, pertimbangkan varian dan fleksibilitas keluarga. Setiap pendekatan memiliki pengorbanan yang bergantung pada kasus penggunaan dan persyaratan Anda.

- Fleksibilitas ukuran — Pertama, diversifikasi dengan contoh ukuran berbeda dalam keluarga yang sama. Instans dalam keluarga yang sama memberikan biaya dan kinerja yang sama, tetapi dapat meluncurkan jumlah kontainer yang berbeda di setiap host. Misalnya, jika ukuran eksekutor minimum yang Anda butuhkan adalah memori 2vCPU dan 8Gb, ukuran instans minimum adalah. `m5.xlarge` Untuk fleksibilitas ukuran, sertakan `m5.xlarge``m5.2xlarge`,`m5.4xlarge`,`m5.8xlarge`,`m5.12xlarge`,,`m5.16xlarge`, dan`m5.24xlarge`.
- Fleksibilitas Graviton — Selain ukuran, Anda dapat melakukan diversifikasi dengan instance Graviton. Instans Graviton didukung oleh prosesor AWS Graviton2 yang memberikan kinerja harga terbaik untuk beban kerja cloud di Amazon EC2. Misalnya, dengan ukuran instans minimum`m5.xlarge`, Anda dapat menyertakan,,`m6g.xlarge`,, `m6g.2xlarge` `m6g.4xlarge``m6g.8xlarge`, dan `m6g.16xlarge` untuk fleksibilitas Graviton.
- Fleksibilitas generasi — Mirip dengan Graviton dan fleksibilitas ukuran, instance dalam keluarga generasi sebelumnya memiliki spesifikasi perangkat keras yang sama. Ini menghasilkan profil biaya dan kinerja yang serupa dengan peningkatan total kumpulan Amazon EC2 yang dapat diakses. Untuk fleksibilitas generasi, sertakan `m4.xlarge``m4.2xlarge`,,`m4.10xlarge`, dan`m4.16xlarge`.
- Fleksibilitas keluarga dan varian
  - Kapasitas — Untuk mengoptimalkan kapasitas, kami merekomendasikan fleksibilitas instans di seluruh keluarga instans. Contoh umum dari keluarga instance yang berbeda memiliki kumpulan contoh yang lebih dalam yang dapat membantu memenuhi persyaratan kapasitas. Namun, instance dari keluarga yang berbeda akan memiliki rasio vCPU terhadap memori yang berbeda. Ini menghasilkan pemanfaatan yang kurang jika wadah aplikasi yang diharapkan berukuran untuk instance yang berbeda. Misalnya, dengan`m5.xlarge`, sertakan instance yang dioptimalkan komputasi seperti `c5` atau instance yang dioptimalkan memori seperti misalnya fleksibilitas keluarga. `r5`
  - Biaya — Untuk mengoptimalkan biaya, kami merekomendasikan fleksibilitas instans di seluruh varian. Instans ini memiliki rasio memori dan vCPU yang sama dengan instance awal.

Pertukaran dengan fleksibilitas varian adalah bahwa kasus ini memiliki kumpulan kapasitas yang lebih kecil yang dapat mengakibatkan kapasitas tambahan yang terbatas atau gangguan Spot yang lebih tinggi. `m5.xlarge`Misalnya, sertakan instance berbasis AMD (`m5a`), instance berbasis SSD () atau instans yang dioptimalkan jaringan (`m5d`) misalnya fleksibilitas varian. `m5n`

Praktik terbaik untuk konfigurasi klaster

Gunakan panduan di bagian ini untuk membantu Anda menentukan tipe instans, opsi pembelian, dan jumlah penyimpanan yang akan disediakan untuk setiap jenis simpul dalam klaster EMR.

Tipe instans apa yang harus Anda gunakan?

Ada beberapa cara untuk menambahkan instans Amazon EC2 ke cluster. Metode yang harus Anda pilih bergantung pada apakah Anda menggunakan konfigurasi grup instance atau konfigurasi armada instance untuk cluster.

- Grup Instance
  - Tambahkan instans dengan tipe yang sama secara manual ke grup instans inti dan tugas yang ada.
  - Tambahkan grup instans tugas secara manual, yang dapat menggunakan tipe instans yang berbeda.
  - Siapkan penskalaan otomatis di Amazon EMR untuk grup instans, menambahkan dan menghapus instance secara otomatis berdasarkan nilai metrik CloudWatch Amazon yang Anda tentukan. Untuk informasi selengkapnya, lihat Gunakan penskalaan cluster.
- Armada Instance
  - Tambahkan satu armada instans tugas.
  - Ubah kapasitas target untuk Instans Sesuai Permintaan dan Spot untuk armada instans inti dan tugas yang ada. Untuk informasi selengkapnya, lihat Mengkonfigurasi armada instans.

Salah satu cara untuk merencanakan instans klaster Anda adalah dengan menjalankan klaster uji dengan kumpulan sampel data yang representatif dan memantau pemanfaatan simpul dalam klaster. Untuk informasi selengkapnya, lihat Melihat dan memantau suatu klaster. Cara lain adalah dengan menghitung kapasitas instans yang Anda pertimbangkan dan membandingkan nilai tersebut dengan ukuran data Anda.

Secara umum, tipe node utama, yang menetapkan tugas, tidak memerlukan instans EC2 dengan banyak daya pemrosesan; Instans Amazon EC2 untuk tipe node inti, yang memproses tugas

dan menyimpan data dalam HDFS, memerlukan daya pemrosesan dan kapasitas penyimpanan; Instans Amazon EC2 untuk tipe node tugas, yang tidak menyimpan data, hanya membutuhkan daya pemrosesan. Untuk panduan tentang instans Amazon EC2 yang tersedia dan konfigurasinya, lihat. Konfigurasikan instans Amazon EC2

Panduan berikut berlaku untuk sebagian besar klaster Amazon EMR.

- Ada batas vCPU untuk jumlah total instans Amazon EC2 sesuai permintaan yang Anda jalankan pada akun per. AWS Wilayah AWS Untuk informasi selengkapnya tentang batas vCPU dan cara meminta peningkatan batas untuk akun Anda, lihat Instans Sesuai Permintaan di Panduan Pengguna Amazon EC2 untuk Instans Linux.
- Node primer biasanya tidak memiliki persyaratan komputasi yang besar. Untuk cluster dengan sejumlah besar node, atau untuk cluster dengan aplikasi yang secara khusus digunakan pada node primer (JupyterHub, Hue, dll.), node primer yang lebih besar mungkin diperlukan dan dapat membantu meningkatkan kinerja cluster. Misalnya, pertimbangkan untuk menggunakan instans m5.xlarge untuk klaster yang berukuran kecil (50 simpul atau lebih sedikit), dan tingkatkan ke tipe instans yang lebih besar untuk klaster yang lebih besar.
- Kebutuhan komputasi dari simpul inti dan tugas bergantung pada jenis pemrosesan yang dilakukan aplikasi Anda. Berbagai pekerjaan dapat dijalankan pada tipe instans tujuan umum, yang menawarkan performa seimbang dalam hal CPU, ruang disk, dan input/output. Klaster intensif komputasi dapat mengambil manfaat dari menjalankan instans CPU Tinggi, yang memiliki CPU lebih banyak secara proporsional daripada RAM. Aplikasi basis data dan cache memori dapat mengambil manfaat dari menjalankan instans Memori Tinggi. Aplikasi intensif jaringan dan intensif CPU seperti parsing, NLP, dan machine learning dapat mengambil manfaat dari menjalankan pada instans komputasi klaster, yang menyediakan sumber daya CPU tinggi secara proporsional dan peningkatan performa jaringan.
- Jika fase yang berbeda dari klaster Anda memiliki kebutuhan kapasitas yang berbeda, Anda dapat memulai dengan sejumlah kecil simpul inti dan menambah atau mengurangi jumlah simpul tugas untuk memenuhi berbagai persyaratan kapasitas alur kerja Anda.
- Jumlah data yang dapat Anda proses bergantung pada kapasitas simpul inti dan ukuran data Anda sebagai input, selama pemrosesan, dan sebagai output. Set data input, menengah, dan output semuanya berada di klaster selama pemrosesan.

Kapan Anda harus menggunakan Instans Spot?

Saat meluncurkan klaster di Amazon EMR, Anda dapat memilih untuk meluncurkan instance utama, inti, atau tugas di Instans Spot. Karena setiap jenis grup instans memainkan peran yang berbeda

dalam klaster, terdapat implikasi peluncuran dari setiap jenis simpul pada Instans Spot. Anda tidak dapat mengubah opsi pembelian instans saat klaster sedang berjalan. Untuk mengubah dari On-Demand ke Instans Spot atau sebaliknya, untuk node primer dan inti, Anda harus menghentikan cluster dan meluncurkan yang baru. Untuk simpul tugas, Anda dapat meluncurkan grup instans tugas atau armada instans baru, dan menghapus yang lama.

Topik

Pengaturan Amazon EMR untuk mencegah kegagalan tugas karena pengakhiran Instans Spot simpul tugas

Karena Instans Spot sering digunakan untuk menjalankan simpul tugas, Amazon EMR memiliki fungsionalitas default untuk menjadwalkan tugas YARN sehingga tugas yang sedang berjalan tidak mengalami kegagalan saat simpul tugas yang berjalan pada Instans Spot diakhiri. Amazon EMR melakukan ini dengan mengizinkan proses utama aplikasi berjalan hanya pada simpul inti. Proses utama aplikasi mengontrol tugas yang sedang berjalan dan harus tetap hidup selama masa tugas.

Amazon EMR merilis 5.19.0 dan yang lebih baru menggunakan fitur label node YARN bawaan untuk mencapai ini. (Versi sebelumnya menggunakan patch kode). Properti dalam klasifikasi konfigurasi `yarn-site` dan `capacity-scheduler` dikonfigurasi secara default sehingga YARN capacity-scheduler dan fair-scheduler memanfaatkan label simpul. Amazon EMR secara otomatis melabeli simpul inti dengan label `CORE`, dan menetapkan properti sehingga utama aplikasi dijadwalkan hanya pada simpul dengan label INTI. Mengubah properti terkait secara manual dalam klasifikasi konfigurasi yarn-site dan capacity-scheduler, atau secara langsung dalam file XML terkait, dapat merusak fitur ini atau mengubah fungsionalitas ini.

Amazon EMR mengonfigurasi properti dan nilai berikut secara default. Berhati-hatilah saat mengonfigurasi properti ini.

- yarn-site (yarn-site.xml) Pada Semua Node
  - `yarn.node-labels.enabled: true`

  - `yarn.node-labels.am.default-node-label-expression: 'CORE'`
  - `yarn.node-labels.fs-store.root-dir: '/apps/yarn/nodelabels'`
  - `yarn.node-labels.configuration-type: 'distributed'`
- yarn-site (yarn-site.xml) Pada Node Primer Dan Inti
  - `yarn.nodemanager.node-labels.provider: 'config'`
  - `yarn.nodemanager.node-labels.provider.configured-node-partition: 'CORE'`
- capacity-scheduler (capacity-scheduler.xml) Pada Semua Node
  - `yarn.scheduler.capacity.root.accessible-node-labels: '*'`
  - `yarn.scheduler.capacity.root.accessible-node-labels.CORE.capacity: 100`
  - `yarn.scheduler.capacity.root.default.accessible-node-labels: '*'`
  - `yarn.scheduler.capacity.root.default.accessible-node-labels.CORE.capacity: 100`

> **Note**
>
> Dimulai dengan Amazon EMR seri rilis 6.x, fitur label simpul YARN dinonaktifkan secara default. Proses utama aplikasi dapat berjalan pada node inti dan tugas secara default. Anda dapat mengaktifkan fitur label simpul YARN dengan mengkonfigurasi properti berikut:
>
> - `yarn.node-labels.enabled: true`
> - `yarn.node-labels.am.default-node-label-expression: 'CORE'`

Node utama pada Instance Spot

Node utama mengontrol dan mengarahkan cluster. Ketika berakhir, cluster berakhir, jadi Anda hanya harus meluncurkan node utama sebagai Instans Spot jika Anda menjalankan cluster di mana penghentian mendadak dapat diterima. Ini mungkin terjadi jika Anda menguji aplikasi baru, memiliki klaster yang secara berkala menyimpan data ke penyimpanan eksternal seperti Amazon S3, atau menjalankan klaster di mana biaya lebih penting daripada memastikan penyelesaian klaster.

Saat Anda meluncurkan grup instans utama sebagai Instans Spot, klaster tidak akan dimulai hingga permintaan Instans Spot terpenuhi. Ini adalah sesuatu yang perlu dipertimbangkan ketika memilih harga Spot maksimum Anda.

Anda hanya dapat menambahkan node utama Instance Spot saat meluncurkan cluster. Anda tidak dapat menambah atau menghapus node utama dari cluster yang sedang berjalan.

Biasanya, Anda hanya akan menjalankan node utama sebagai Instance Spot jika Anda menjalankan seluruh cluster (semua grup instance) sebagai Instans Spot.

Simpul inti pada Instans Spot

Simpul inti memproses data dan menyimpan informasi menggunakan HDFS. Mengakhiri instans inti mengakibatkan risiko kehilangan data. Karena alasan ini, Anda hanya boleh menjalankan simpul inti pada Instans Spot jika kehilangan sebagian data HDFS dapat ditoleransi.

Saat Anda meluncurkan grup instans inti sebagai Instans Spot, Amazon EMR menunggu hingga dapat menyediakan semua instans inti yang diminta sebelum meluncurkan grup instans. Dengan kata lain, jika Anda meminta enam instans Amazon EC2, dan hanya lima yang tersedia pada atau di bawah harga Spot maksimum Anda, maka grup instans tidak akan diluncurkan. Amazon EMR terus menunggu hingga keenam instans Amazon EC2 tersedia atau hingga Anda mengakhiri klaster. Anda dapat mengubah jumlah Instans Spot dalam grup instans inti untuk menambah kapasitas ke klaster yang sedang berjalan. Untuk informasi selengkapnya tentang bekerja dengan grup instans, dan bagaimana Instans Spot bekerja dengan armada instans, lihat the section called “Konfigurasikan armada instans atau grup instans”.

Simpul tugas pada Instans Spot

Simpul tugas memproses data tetapi tidak menyimpan data persisten dalam HDFS. Jika mereka berakhir karena harga Spot telah naik di atas harga Spot maksimum Anda, tidak ada data yang hilang dan hanya akan terjadi efek minim pada klaster Anda.

Saat Anda meluncurkan satu atau beberapa grup instans tugas sebagai Instans Spot, Amazon EMR menyediakan simpul tugas sebanyak mungkin, menggunakan harga Spot maksimum Anda. Ini berarti bahwa jika Anda meminta grup instans tugas dengan enam simpul, dan hanya lima Instans Spot yang tersedia pada atau di bawah harga Spot maksimum Anda, Amazon EMR akan meluncurkan grup instans dengan lima simpul lalu menambahkan yang keenam nanti jika memungkinkan.

Meluncurkan grup instans tugas sebagai Instans Spot adalah cara strategis untuk memperluas kapasitas klaster Anda sekaligus meminimalkan biaya. Jika Anda meluncurkan grup instans utama dan inti sebagai Instans Sesuai Permintaan, kapasitasnya dijamin untuk menjalankan klaster. Anda dapat menambahkan instans tugas ke grup instans tugas sesuai kebutuhan, untuk menangani lalu lintas puncak atau mempercepat pemrosesan data.

Anda dapat menambah atau menghapus simpul tugas menggunakan konsol, AWS CLI, atau API. Anda juga dapat menambahkan grup tugas tambahan, tetapi Anda tidak dapat menghapus grup tugas setelah dibuat.

Konfigurasi instans untuk skenario aplikasi

Tabel berikut adalah referensi cepat untuk opsi dan konfigurasi pembelian tipe simpul yang biasanya sesuai untuk berbagai skenario aplikasi. Pilih tautan untuk melihat informasi selengkapnya tentang setiap jenis skenario.

Skenario aplikasi	Opsi pembelian simpul utama	Opsi pembelian simpul inti	Opsi pembelian simpul tugas
klaster dan gudang data yang berjalan lama	Sesuai Permintaan	Gabungan Sesuai Permintaan atau armada instans	Gabungan spot atau armada instans
Beban kerja dengan biaya	Spot	Spot	Spot
Beban kerja data kritis	Sesuai Permintaan	Sesuai Permintaan	Gabungan spot atau armada instans
Pengujian aplikasi	Spot	Spot	Spot

Ada beberapa skenario di mana Instans Spot berguna untuk menjalankan klaster Amazon EMR.

klaster dan gudang data yang berjalan lama

Jika Anda menjalankan klaster Amazon EMR persisten yang memiliki variasi kapasitas komputasi yang dapat diprediksi, seperti gudang data, Anda dapat menangani permintaan puncak dengan biaya lebih rendah menggunakan Instans Spot. Anda dapat meluncurkan grup instans utama dan inti sebagai Instans Sesuai Permintaan untuk menangani kapasitas normal dan meluncurkan grup instans tugas sebagai Instans Spot untuk menangani persyaratan beban puncak Anda.

Beban kerja dengan biaya

Jika Anda menjalankan klaster sementara yang biayanya lebih rendah lebih penting daripada waktu penyelesaian, dan kehilangan sebagian pekerjaan dapat diterima, Anda dapat menjalankan seluruh

klaster (grup instance primer, inti, dan tugas) sebagai Instans Spot untuk mendapatkan keuntungan dari penghematan biaya terbesar.

Beban kerja data kritis

Jika Anda menjalankan klaster yang biayanya lebih rendah lebih penting daripada waktu penyelesaian, tetapi kehilangan sebagian pekerjaan tidak dapat diterima, luncurkan grup instans utama dan inti sebagai Instans Sesuai Permintaan dan lengkapi dengan satu atau beberapa grup instans tugas dari Instans Spot. Menjalankan grup instans utama dan inti sebagai Instans Sesuai Permintaan memastikan bahwa data Anda disimpan dalam HDFS dan klaster terlindungi dari penghentian karena fluktuasi pasar Spot, sekaligus memberikan penghematan biaya yang timbul dari menjalankan grup instans tugas sebagai Instans Spot.

Pengujian aplikasi

Saat Anda menguji aplikasi baru untuk mempersiapkannya untuk diluncurkan di lingkungan produksi, Anda dapat menjalankan seluruh cluster (grup instance utama, inti, dan tugas) sebagai Instans Spot untuk mengurangi biaya pengujian Anda.

Menghitung kapasitas HDFS yang dibutuhkan dari sebuah klaster

Jumlah penyimpanan HDFS yang tersedia untuk cluster Anda tergantung pada faktor-faktor berikut:

- Jumlah instans Amazon EC2 yang digunakan untuk node inti.
- Kapasitas penyimpanan instans Amazon EC2 untuk jenis instans yang digunakan. Untuk informasi selengkapnya tentang volume penyimpanan instans, lihat penyimpanan instans Amazon EC2 Amazon di Panduan Pengguna Amazon EC2 untuk Instans Linux.
- Jumlah dan ukuran volume Amazon EBS yang melekat pada node inti.
- Faktor replikasi, yang mana menjelaskan bagaimana setiap blok data disimpan dalam HDFS untuk redundansi seperti RAID. Secara default, faktor replikasi adalah tiga untuk sebuah klaster yang memiliki 10 atau lebih simpul inti, dua untuk sebuah klaster yang memiliki 4-9 simpul inti, dan satu untuk satu kalster yang memiliki tiga atau lebih sedikit simpul.

Untuk menghitung kapasitas HDFS sebuah cluster, untuk setiap node inti, tambahkan kapasitas volume penyimpanan instans ke kapasitas penyimpanan Amazon EBS (jika digunakan). Kalikan hasilnya dengan jumlah simpul inti, lalu bagi total dengan faktor replikasi berdasarkan jumlah simpul inti. Misalnya, sebuah cluster dengan 10 node inti tipe i2.xlarge, yang memiliki penyimpanan instans 800 GB tanpa volume Amazon EBS yang terpasang, memiliki total sekitar 2.666 GB yang tersedia untuk HDFS (10 node x 800 GB ÷ 3 faktor replikasi).

Jika nilai kapasitas HDFS yang dihitung lebih kecil dari data Anda, Anda dapat menambah jumlah penyimpanan HDFS dengan cara berikut:

- Membuat klaster dengan volume Amazon EBS tambahan atau menambahkan grup instans dengan volume Amazon EBS terlampir ke klaster yang ada
- Menambahkan lebih banyak simpul inti
- Memilih jenis instans Amazon EC2 dengan kapasitas penyimpanan yang lebih besar
- Menggunakan kompresi data
- Mengubah pengaturan konfigurasi Hadoop untuk mengurangi faktor replikasi

Mengurangi faktor replikasi harus digunakan dengan hati-hati karena dapat mengurangi redundansi data HDFS dan kemampuan klaster untuk memulihkan dari blok HDFS yang hilang atau rusak.

Konfigurasi pencatatan log dan debugging klaster

Salah satu hal yang harus diputuskan saat Anda merencanakan klaster adalah seberapa banyak dukungan debugging yang ingin Anda sediakan. Saat pertama kali mengembangkan aplikasi pemrosesan data, sebaiknya uji aplikasi pada klaster yang memproses sebagian kecil, namun mewakili data Anda. Jika Anda melakukan ini, Anda mungkin ingin memanfaatkan semua alat debugging yang ditawarkan Amazon EMR, seperti pengarsipan berkas log ke Amazon S3.

Setelah Anda menyelesaikan pengembangan dan memasukkan aplikasi pemrosesan data ke produksi penuh, Anda dapat memilih untuk mengurangi skala debugging. Melakukannya dapat menghemat biaya penyimpanan arsip berkas log di Amazon S3 dan mengurangi beban pemrosesan pada klaster karena tidak perlu lagi menulis status ke Amazon S3. Keuntungannya, tentu saja, adalah jika terjadi kesalahan, Anda hanya membutuhkan lebih sedikit alat untuk menyelidiki masalah tersebut.

berkas log default

Secara default, setiap cluster menulis file log pada node utama. Ini ditulis untuk direktori `/mnt/var/log/`. Anda dapat mengaksesnya dengan menggunakan SSH untuk terhubung ke node utama seperti yang dijelaskan dalamConnect ke node utama menggunakan SSH.

> **Note**
>
> Jika Anda menggunakan Amazon EMR release 6.8.0 atau versi lebih lama, file log disimpan ke Amazon S3 selama penghentian klaster, sehingga Anda tidak dapat mengakses file log setelah node utama berakhir. Amazon EMR merilis 6.9.0 dan log arsip yang lebih baru ke Amazon S3 selama penskalaan cluster, sehingga file log yang dihasilkan di cluster tetap ada bahkan setelah node dihentikan.

Anda tidak perlu mengaktifkan apa pun untuk memiliki file log yang ditulis pada simpul utama. Ini adalah perilaku default Amazon EMR dan Hadoop.

Sebuah klaster menghasilkan beberapa jenis berkas log, termasuk:

- Langkah log — Log ini dihasilkan oleh layanan Amazon EMR dan berisi informasi tentang klaster dan hasil dari setiap langkah. File log disimpan dalam `/mnt/var/log/hadoop/steps/` direktori pada node utama. Setiap langkah mencatat hasilnya dalam subdirektori bernomor terpisah: `/mnt/var/log/hadoop/steps/s-stepId1/` untuk langkah pertama, `/mnt/var/log/hadoop/steps/s-stepId2/`, untuk langkah kedua, dan seterusnya. Pengidentifikasi langkah 13 karakter (misalnya stepId1, stepId2) unik untuk sebuah klaster.
- Log komponen Hadoop dan YARN — Log untuk komponen yang terkait dengan Apache YARN dan MapReduce, misalnya, terkandung dalam folder terpisah di. `/mnt/var/log` Lokasi berkas log untuk komponen Hadoop di bawah `/mnt/var/log` adalah sebagai berikut: hadoop-hdfs, hadoop-mapreduce, hadoop-httpfs, dan hadoop-yarn. hadoop-state-pusher Direktori adalah untuk output dari proses pendorong status Hadoop.
- Log tindakan bootstrap — Jika tugas Anda menggunakan tindakan bootstrap, hasil dari tindakan tersebut akan dicatat. File log disimpan di /mnt/var/log/bootstrap-actions/ pada node utama. Setiap tindakan bootstrap mencatat hasilnya di subdirektori bernomor terpisah: `/mnt/var/log/bootstrap-actions/1/` untuk tindakan bootstrap pertama, `/mnt/var/log/bootstrap-actions/2/`, untuk tindakan bootstrap kedua, dan seterusnya.
- Log status instans- Log ini memberikan informasi tentang CPU, status memori, dan utas pengumpul sampah dari simpul. File log disimpan `/mnt/var/log/instance-state/` di simpul utama.

Arsipkan berkas log ke Amazon S3

> **Note**
>
> Saat ini Anda tidak dapat menggunakan agregasi log ke Amazon S3 dengan utilitas `yarn logs`.

Amazon EMR merilis 6.9.0 dan log arsip yang lebih baru ke Amazon S3 selama penskalaan cluster, sehingga file log yang dihasilkan di cluster tetap ada bahkan setelah node dihentikan. Perilaku ini diaktifkan secara otomatis, jadi Anda tidak perlu melakukan apa pun untuk menyalakannya. Untuk Amazon EMR rilis 6.8.0 dan versi lebih lama, Anda dapat mengonfigurasi cluster untuk mengarsipkan file log yang disimpan di node utama ke Amazon S3 secara berkala. Hal ini memastikan bahwa berkas log tersedia setelah klaster berakhir, baik melalui pematian normal atau karena kesalahan. Amazon EMR arsip berkas log ke Amazon S3 dengan interval 5 menit.

Agar file log diarsipkan ke Amazon S3 untuk Amazon EMR rilis 6.8.0 dan yang lebih lama, Anda harus mengaktifkan fitur ini saat meluncurkan cluster. Anda dapat melakukannya dengan menggunakan konsol, CLI, atau API. Secara default, klaster diluncurkan dengan menggunakan konsol yang telah mengaktifkan pengarsipan log. Untuk klaster yang diluncurkan menggunakan CLI atau API, log ke Amazon S3 harus diaktifkan secara manual.

> **Note**
>
> Kami telah mendesain ulang konsol EMR Amazon agar lebih mudah digunakan. Lihat Apa yang baru dengan konsol? untuk mempelajari tentang perbedaan antara pengalaman konsol lama dan baru.

New console

Untuk mengarsipkan file log ke Amazon S3 dengan konsol baru

1. Masuk keAWS Management Console, dan buka konsol EMR Amazon di https://console.aws.amazon.com/emr.
2. Di bawah EMR pada EC2 di panel navigasi kiri, pilih Clusters, lalu pilih Create cluster.
3. Di bawah Log klaster, pilih kotak centang Publikasikan log khusus klaster ke Amazon S3.

4. Di bidang lokasi Amazon S3, ketik (atau telusuri ke) jalur Amazon S3 untuk menyimpan log Anda. Jika Anda mengetik nama folder yang tidak ada di bucket, Amazon S3 membuatnya.

   Saat Anda menetapkan nilai ini, Amazon EMR menyalin file log dari instans EC2 di cluster ke Amazon S3. Ini mencegah file log hilang saat cluster berakhir dan EC2 menghentikan instance yang menghosting cluster. Log ini berguna untuk tujuan pemecahan masalah. Untuk informasi lebih lanjut tentang format berkas log, lihat Tampilkan berkas log.
5. Secara opsional, pilih kotak centang Encrypt cluster-specific logs. Kemudian, pilih AWS KMS kunci dari daftar, masukkan kunci ARN, atau buat kunci baru. Opsi ini hanya tersedia dengan Amazon EMR versi 5.30.0 dan yang lebih baru, tidak termasuk versi 6.0.0. Untuk menggunakan opsi ini, tambahkan izin AWS KMS untuk profil instans EC2 Anda dan peran Amazon EMR. Untuk informasi selengkapnya, lihat Untuk mengenkripsi berkas log yang disimpan di Amazon S3 dengan AWS kunci yang dikelola pelanggan KMS.
6. Pilih opsi lain yang berlaku untuk cluster Anda.
7. Untuk meluncurkan klaster Anda, pilih Buat klaster.

Old console

Untuk mengarsipkan file log ke Amazon S3 dengan konsol lama

1. Arahkan ke konsol EMR Amazon baru dan pilih Beralih ke konsol lama dari navigasi samping. Untuk informasi selengkapnya tentang apa yang diharapkan saat beralih ke konsol lama, lihat Menggunakan konsol lama.
2. Pilih Buat klaster.
3. Pilih Pergi ke opsi lanjutan.
4. Di bagian Opsi umum, di bidang Pencatatan log, terima opsi default: Diaktifkan.

   Hal ini menentukan apakah Amazon EMR menangkap data log terperinci ke Amazon S3. Anda hanya dapat mengatur ini ketika klaster dibuat. Untuk informasi selengkapnya, lihat Melihat berkas log.
5. Di bidang Folder S3, ketik (atau jelajahi ke) jalur Amazon S3 untuk menyimpan log Anda. Anda juga dapat mengizinkan konsol untuk menghasilkan jalur Amazon S3 untuk Anda. Jika Anda mengetik nama folder yang tidak ada di bucket, nama folder tersebut akan dibuat.

Saat nilai ini ditetapkan, Amazon EMR menyalin berkas log dari instans EC2 di klaster ke Amazon S3. Hal ini mencegah berkas log hilang saat klaster berakhir dan instans EC2 yang menghosting klaster diakhiri. Log ini berguna untuk tujuan pemecahan masalah.

Untuk informasi lebih lanjut tentang format berkas log, lihat Tampilkan berkas log.

6. Di bidang Enkripsi log, pilih Enkripsi log yang disimpan di S3 dengan AWS Kunci yang dikelola pelanggan KMS. Kemudian pilih AWS Kunci KMS dari daftar atau masukkan ARN kunci. Anda juga dapat membuat AWS KMS kunci baru.

   Opsi ini hanya tersedia dengan Amazon EMR versi 5.30.0 dan yang lebih baru, tidak termasuk versi 6.0.0. Untuk menggunakan opsi ini, tambahkan izin AWS KMS untuk profil instans EC2 Anda dan peran Amazon EMR. Untuk informasi selengkapnya, lihat Untuk mengenkripsi berkas log yang disimpan di Amazon S3 dengan AWS kunci yang dikelola pelanggan KMS.
7. Lanjutkan dengan membuat klaster seperti yang dijelaskan dalam Merencanakan dan mengonfigurasi klaster.

CLI

Untuk mengarsipkan file log ke Amazon S3 dengan AWS CLI

Untuk mengarsipkan berkas log ke Amazon S3 menggunakan AWS CLI, ketik perintah `create-cluster` dan tentukan jalur log Amazon S3 menggunakan parameter `--log-uri`.

1. Untuk berkas log ke Amazon S3 ketik perintah berikut dan ganti *myKey* dengan nama pasangan kunci EC2 Anda.

```
aws emr create-cluster --name "Test cluster" --release-label emr-7.0.0 --log-
uri s3://DOC-EXAMPLE-BUCKET/logs --applications Name=Hadoop Name=Hive Name=Pig
 --use-default-roles --ec2-attributes KeyName=myKey --instance-type m5.xlarge --
instance-count 3
```

2. Saat Anda menentukan jumlah instance tanpa menggunakan `--instance-groups` parameter, satu node primer diluncurkan, dan instance yang tersisa diluncurkan sebagai node inti. Semua simpul akan menggunakan tipe instans yang ditentukan dalam perintah.

> **Note**
>
> Jika sebelumnya Anda belum membuat peran layanan EMR Amazon default dan profil instans EC2, masukkan `aws emr create-default-roles` untuk membuatnya sebelum mengetik subperintah. `create-cluster`

Untuk mengenkripsi berkas log yang disimpan di Amazon S3 dengan AWS kunci yang dikelola pelanggan KMS

Dengan Amazon EMR versi 5.30.0 dan yang lebih baru (kecuali Amazon EMR 6.0.0), Anda dapat mengenkripsi berkas log yang disimpan di Amazon S3 dengan AWS kunci yang dikelola pelanggan KMS. Untuk mengaktifkan opsi ini di konsol, ikuti langkah-langkah di Arsipkan berkas log ke Amazon S3. Profil instans Amazon EC2 dan peran Amazon EMR Anda harus memenuhi prasyarat berikut:

- Profil instans Amazon EC2 yang digunakan untuk klaster Anda harus memiliki izin untuk menggunakan `kms:GenerateDataKey`.
- Peran Amazon EMR yang digunakan untuk klaster Anda harus memiliki izin untuk menggunakan `kms:DescribeKey`.
- Profil instans Amazon EC2 dan peran Amazon EMR harus ditambahkan ke daftar pengguna kunci untuk AWS kunci yang dikelola pelanggan KMS yang ditentukan, seperti yang dijelaskan melalui langkah-langkah berikut:
  1. Buka konsol AWS Key Management Service (AWS KMS) di https://console.aws.amazon.com/kms.
  2. Untuk mengubah Wilayah AWS, gunakan pemilih Wilayah di sudut kanan atas halaman.
  3. Pilih alias tombol KMS untuk memodifikasi.
  4. Pada halaman detail kunci di bawah Pengguna Kunci, pilih Tambahkan.
  5. Di kotak dialog Tambahkan pengguna kunci, pilih profil instans Amazon EC2 dan peran Amazon EMR.
  6. Pilih Tambahkan.

Untuk informasi selengkapnya, lihat Peran layanan IAM yang digunakan oleh Amazon EMR, dan Menggunakan kebijakan kunci dalam AWS Panduan developer Layanan Manajemen Kunci.

Untuk menggabungkan log di Amazon S3 menggunakan AWS CLI

> **Note**
>
> Saat ini Anda tidak dapat menggunakan agregasi log dengan utilitas `yarn logs`. Anda hanya dapat menggunakan agregasi yang didukung oleh prosedur ini.

Agregasi log (Hadoop 2.x) mengkompilasi log dari semua kontainer untuk aplikasi individual ke dalam satu file. Untuk mengaktifkan agregasi log ke Amazon S3 menggunakan AWS CLI, Anda menggunakan tindakan bootstrap saat peluncuran klaster untuk mengaktifkan agregasi log dan guna menentukan bucket untuk menyimpan log.

- Untuk mengaktifkan agregasi log, buat file konfigurasi berikut `myConfig.json` yang disebut yang berisi berikut ini:

```
[
  {
    "Classification": "yarn-site",
    "Properties": {
      "yarn.log-aggregation-enable": "true",
      "yarn.log-aggregation.retain-seconds": "-1",
      "yarn.nodemanager.remote-app-log-dir": "s3:\/\/DOC-EXAMPLE-BUCKET\/logs"
    }
  }
]
```

  Ketik perintah berikut dan ganti *myKey*dengan nama key pair EC2 Anda. Anda juga dapat mengganti salah satu teks merah dengan konfigurasi Anda sendiri.

```
aws emr create-cluster --name "Test cluster" \
--release-label emr-7.0.0 \
--applications Name=Hadoop \
--use-default-roles \
--ec2-attributes KeyName=myKey \
--instance-type m5.xlarge \
--instance-count 3 \
--configurations file://./myConfig.json
```

Saat Anda menentukan jumlah instance tanpa menggunakan `--instance-groups` parameter, satu node primer diluncurkan, dan instance yang tersisa diluncurkan sebagai node inti. Semua simpul akan menggunakan tipe instans yang ditentukan dalam perintah.

> **Note**
>
> Jika sebelumnya Anda belum membuat peran layanan EMR default dan profil instans EC2, jalankan `aws emr create-default-roles` untuk membuatnya sebelum menjalankan subperintah. `create-cluster`

Untuk informasi selengkapnya tentang menggunakan perintah Amazon EMR di AWS CLI, lihat AWS CLI Refensi Perintah.

Log lokasi

Daftar berikut mencakup semua jenis log dan lokasinya di Amazon S3. Anda dapat menggunakan ini untuk memecahkan masalah Amazon EMR.

Log langkah

```
s3://DOC-EXAMPLE-LOG-BUCKET/<cluster-id>/steps/<step-id>/
```

Log aplikasi

```
s3://DOC-EXAMPLE-LOG-BUCKET/<cluster-id>/containers/
```

Lokasi ini termasuk kontainer `stderr` dan `stdoutdirectory.info`,`prelaunch.out`,, dan `launch_container.sh` log.

Log manajer sumber daya

```
s3://DOC-EXAMPLE-LOG-BUCKET/<cluster-id>/node/<leader-instance-id>/
applications/hadoop-yarn/
```

Hadoop HDFS

```
s3://DOC-EXAMPLE-LOG-BUCKET/<cluster-id>/node/<all-instance-id>/
applications/hadoop-hdfs/
```

Lokasi ini termasuk NameNode, DataNode, dan TimelineServer log YARN.

Log manajer simpul

```
s3://DOC-EXAMPLE-LOG-BUCKET/<cluster-id>/node/<all-instance-id>/
applications/hadoop-yarn/
```

Log instance-state

```
s3://DOC-EXAMPLE-LOG-BUCKET/<cluster-id>/node/<all-instance-id>/daemons/
instance-state/
```

Log penyediaan EMR Amazon

```
s3://DOC-EXAMPLE-LOG-BUCKET/<cluster-id>/node/<leader-instance-id>/
provision-node/*
```

Log sarang

```
s3://DOC-EXAMPLE-LOG-BUCKET/<cluster-id>/node/<leader-instance-id>/
applications/hive/*
```

- Untuk menemukan log Hive di cluster Anda, hapus tanda bintang (*) dan tambahkan `/var/log/hive/` ke tautan di atas.
- Untuk menemukan HiveServer 2 log, hapus tanda bintang (*) dan tambahkan `var/log/hive/hiveserver2.log` ke tautan di atas.
- Untuk menemukan log HiveCli, hapus tanda bintang (*) dan tambahkan `/var/log/hive/user/hadoop/hive.log` ke tautan di atas.
- Untuk menemukan log Hive Metastore Server, hapus tanda bintang (*) dan tambahkan ke tautan di atas. `/var/log/hive/user/hive/hive.log`

Jika kegagalan Anda berada di simpul utama atau tugas aplikasi Tez Anda, berikan log dari wadah Hadoop yang sesuai.

Aktifkan alat debugging

Alat debugging memungkinkan Anda untuk lebih mudah menelusuri file log dari konsol EMR Amazon. Untuk informasi selengkapnya, lihat Melihat berkas log dalam alat debugging. Saat Anda mengaktifkan debugging pada klaster, Amazon EMR mengarsipkan berkas log ke Amazon S3 lalu mengindeks file tersebut. Anda kemudian dapat menggunakan konsol ini untuk menelusuri langkah, pekerjaan, tugas, dan log upaya tugas untuk klaster dengan cara yang intuitif.

Untuk menggunakan alat debugging di konsol EMR Amazon, Anda harus mengaktifkan debugging saat meluncurkan cluster menggunakan konsol, CLI, atau API. Perhatikan bahwa konsol EMR Amazon baru tidak menawarkan alat debugging.

Old console

Untuk mengaktifkan alat debugging dengan konsol lama

1. Arahkan ke konsol EMR Amazon baru dan pilih Beralih ke konsol lama dari navigasi samping. Untuk informasi selengkapnya tentang apa yang diharapkan saat beralih ke konsol lama, lihat Menggunakan konsol lama.
2. Pilih Buat klaster.
3. Pilih Pergi ke opsi lanjutan.
4. Di bagian Konfigurasi klaster, di bidang Pencatatan log, pilih Diaktifkan. Anda tidak dapat mengaktifkan debugging tanpa mengaktifkan pencatatan log.
5. Di bidang Lokasi S3 folder log, ketik jalur Amazon S3 untuk menyimpan log Anda.
6. Di bidang Debugging, pilih Diaktifkan. Opsi debug membuat pertukaran Amazon SQS guna mempublikasikan pesan debug ke backend layanan Amazon EMR. Biaya untuk memublikasikan pesan ke pertukaran mungkin berlaku. Untuk informasi selengkapnya, lihat halaman produk Amazon SQS.
7. Lanjutkan dengan membuat klaster seperti yang dijelaskan dalam Merencanakan dan mengonfigurasi klaster.

AWS CLI

Untuk mengaktifkan alat debugging dengan AWS CLI

Untuk mengaktifkan debugging menggunakan AWS CLI, ketik subperintah `create-cluster` dengan parameter `--enable-debugging`. Anda juga harus menentukan parameter `--log-uri` saat mengaktifkan debugging.

- Untuk mengaktifkan debugging menggunakan AWS CLI, ketik perintah berikut dan ganti *MyKey* dengan nama pasangan kunci EC2 anda.

```
aws emr create-cluster --name "Test cluster" \
--release-label emr-7.0.0 \
--log-uri s3://DOC-EXAMPLE-BUCKET/logs \
--enable-debugging \
```

```
--applications Name=Hadoop Name=Hive Name=Pig \
--use-default-roles \
--ec2-attributes KeyName=myKey \
--instance-type m5.xlarge \
--instance-count 3
```

Saat Anda menentukan jumlah instance tanpa menggunakan `--instance-groups` parameter, satu node primer diluncurkan, dan instance yang tersisa diluncurkan sebagai node inti. Semua simpul akan menggunakan tipe instans yang ditentukan dalam perintah.

> **Note**
>
> Jika Anda belum sebelumnya membuat peran layanan EMR default dan profil instans EC2, ketik aws `emr create-default-roles` untuk membuatnya sebelum mengetik `create-cluster` subperintah.

API

Untuk mengaktifkan alat debugging dengan Amazon EMR API

- Aktifkan debugging menggunakan konfigurasi Java SDK berikut.

```
   StepFactory stepFactory = new StepFactory();
StepConfig enabledebugging = new StepConfig()
    .withName("Enable debugging")
    .withActionOnFailure("TERMINATE_JOB_FLOW")
    .withHadoopJarStep(stepFactory.newEnableDebuggingStep());
```

Dalam contoh ini, `new StepFactory()` menggunakan `us-east-1` sebagai wilayah default. Jika klaster Anda diluncurkan di wilayah yang berbeda, Anda harus menentukan wilayah dengan menggunakan `new StepFactory("region.elasticmapreduce")`, seperti `new StepFactory("ap-northeast-2.elasticmapreduce")`.

Informasi opsi debugging

Amazon EMR merilis 4.1.0 hingga 5.27.0 mendukung debugging di semua Wilayah. Versi EMR Amazon lainnya tidak mendukung opsi debugging. Efektif 23 Januari 2023, Amazon EMR akan menghentikan alat debugging untuk semua versi.

Amazon EMR membuat antrean Amazon SQS untuk memproses data debugging. Biaya pesan mungkin berlaku. Namun, Amazon SQS memiliki Tingkat Gratis hingga 1.000.000 permintaan yang tersedia. Untuk informasi selengkapnya, lihat https://aws.amazon.com/https://aws.amazon.com/sqs.

Debugging memerlukan penggunaan peran; peran layanan dan profil instans Anda harus memungkinkan Anda menggunakan semua operasi API Amazon SQS. Jika peran Anda dilampirkan ke kebijakan terkelola Amazon EMR, Anda tidak perlu melakukan apa pun untuk mengubah peran Anda. Jika Anda memiliki peran kustom, Anda harus menambahkan izin `sqs:*`. Untuk informasi selengkapnya, lihat Konfigurasi peran layanan IAM untuk izin Amazon EMR untuk layanan AWS dan sumber daya.

Klaster tag

Tag dapat memudahkan Anda untuk mengkategorikan sumber daya AWS Anda dengan berbagai cara; misalnya, berdasarkan tujuan, pemilik, atau lingkungan. Anda dapat mencapainya di Amazon EMR dengan menetapkan metadata kustom ke klaster Amazon EMR Anda dengan menggunakan tag. Sebuah tag terdiri atas sebuah kunci dan sebuah nilai, yang keduanya Anda tentukan. Untuk Amazon EMR, klaster adalah tingkat sumber daya yang dapat Anda beri tag. Misalnya, Anda dapat menentukan serangkaian tag untuk klaster akun yang membantu Anda melacak setiap pemilik klaster atau mengidentifikasi klaster produksi versus klaster pengujian. Sebaiknya Anda membuat sekumpulan tag yang sama untuk memenuhi persyaratan organisasi Anda.

Saat Anda menambahkan tag ke klaster Amazon EMR, tag juga disebarkan ke setiap instans Amazon EC2 aktif yang terkait dengan klaster. Demikian pula, saat Anda menghapus tag dari klaster Amazon EMR, tag tersebut akan dihapus dari setiap instans Amazon EC2 aktif terkait.

> **Important**
>
> Gunakan konsol atau CLI Amazon EMR untuk mengelola tag pada instans Amazon EC2 yang merupakan bagian dari klaster alih-alih konsol atau CLI Amazon EC2, karena perubahan yang Anda buat di Amazon EC2 tidak disinkronkan kembali ke sistem penandaan Amazon EMR.

Anda dapat mengidentifikasi instans Amazon EC2 yang merupakan bagian dari klaster Amazon EMR dengan mencari tag sistem berikut. Dalam contoh ini, *`INTI`* adalah nilai untuk peran grup instans dan *`j-12345678`* adalah contoh nilai pengidentifikasi alur kerja (klaster):

- *aws:elasticmapreduce: = INTI instance-group-role*
- *aws:elasticmapreduce: = j-12345678 job-flow-id*

> **Note**
>
> Amazon EMR dan Amazon EC2 menafsirkan tag Anda sebagai string karakter tanpa makna semantik.

Anda dapat bekerja dengan tag menggunakan AWS Management Console, CLI, dan API.

Anda dapat menambahkan tag saat membuat klaster Amazon EMR baru dan Anda juga dapat menambahkan, mengedit, atau menghapus tag dari klaster Amazon EMR yang sedang berjalan. Mengedit tag adalah konsep yang berlaku untuk konsol Amazon EMR, namun menggunakan CLI dan API, untuk mengedit tag Anda akan menghapus tag lama dan menambahkan yang baru. Anda dapat mengedit kunci dan nilai tag, dan Anda dapat menghapus tag dari sumber daya kapan saja klaster berjalan. Namun, Anda tidak dapat menambahkan, mengedit, atau menghapus tag dari klaster yang diakhiri atau instans yang diakhiri yang sebelumnya dikaitkan dengan klaster yang masih aktif. Selain itu, Anda dapat mengatur nilai tag menjadi string kosong, tetapi Anda tidak dapat mengatur nilai tag menjadi nol.

Jika Anda menggunakan AWS Identity and Access Management (IAM) dengan instans Amazon EC2 Anda untuk izin berbasis sumber daya dengan tag, kebijakan IAM Anda akan diterapkan ke tag yang disebarkan Amazon EMR ke instans Amazon EC2 klaster. Agar tag Amazon EMR dapat menyebar ke instans Amazon EC2 Anda, kebijakan IAM Anda untuk Amazon EC2 harus mengizinkan izin untuk memanggil Amazon EC2 dan API. CreateTags DeleteTags Selain itu, tag yang disebarkan dapat memengaruhi izin berbasis sumber daya Amazon EC2 Anda. Tag yang disebarkan ke Amazon EC2 dapat dibaca sebagai syarat dalam kebijakan IAM Anda, sama seperti tag Amazon EC2 lainnya. Ingatlah kebijakan IAM Anda saat menambahkan tag ke kluster EMR Amazon Anda untuk menghindari pengguna memiliki izin yang salah untuk klaster. Untuk menghindari masalah, pastikan bahwa kebijakan IAM Anda tidak menyertakan ketentuan pada tag yang juga Anda rencanakan untuk digunakan pada klaster Amazon EMR Anda. Untuk informasi selengkapnya, lihat Mengontrol akses ke sumber daya Amazon EC2.

Pembatasan tanda

Batasan dasar berikut berlaku untuk tag:

- Pembatasan yang berlaku untuk sumber daya Amazon EC2 berlaku untuk Amazon EMR juga. Untuk informasi selengkapnya, lihat https://docs.aws.amazon.com/AWSEC2/latest/UserGuide/Using_Tags.html#tag-restrictions.
- Jangan menggunakan prefiks `aws:` pada nama atau nilai tag Anda, karena hal ini khusus untuk penggunaan AWS. Selain itu, Anda tidak dapat mengedit atau menghapus nama atau nilai tag dengan prefiks ini.
- Anda tidak dapat mengubah atau mengedit tag pada klaster yang diakhiri.
- Nilai tag dapat berupa string kosong, tetapi bukan nol. Selain itu, kunci tag tidak boleh berupa string kosong.
- Kunci dan nilai dapat berisi karakter alfabet apa pun dalam bahasa apa pun, karakter numerik apa pun, spasi putih, pemisah tak terlihat, dan simbol berikut: _ . : / = + - @

Untuk informasi selengkapnya tentang pemberian tag menggunakan AWS Management Console, lihat Bekerja dengan tag di konsol di Panduan Pengguna Amazon EC2 untuk Instans Linux. Untuk informasi selengkapnya tentang pemberian tag menggunakan Amazon EC2API atau baris perintah, lihat Gambaran umum API dan CLI di Panduan Pengguna Amazon EC2 untuk Instans Linux.

Tag sumber daya untuk penagihan

Anda juga dapat menggunakan tanda untuk mengatur tagihan AWS Anda guna merefleksikan struktur biaya Anda sendiri. Untuk melakukannya, daftar untuk mendapatkan tagihan akun AWS Anda dengan menyertakan nilai kunci tag. Anda kemudian dapat mengatur informasi penagihan berdasarkan nilai kunci tag, untuk melihat biaya sumber daya gabungan Anda. Meskipun Amazon EMR dan Amazon EC2 memiliki laporan penagihan yang berbeda, tag pada setiap klaster juga ditempatkan pada setiap instans terkait sehingga Anda dapat menggunakan tag untuk menautkan biaya Amazon EMR dan Amazon EC2 terkait.

Misalnya, Anda dapat memberi tag pada beberapa sumber daya dengan nama aplikasi tertentu, kemudian mengelola informasi penagihan Anda untuk melihat total biaya aplikasi tersebut di beberapa layanan. Untuk informasi selengkapnya, lihat Tag dan Alokasi Biaya dalam AWS BillingPanduan Pengguna.

Menambahkan tag ke cluster

Anda dapat menambahkan tag ke cluster saat Anda membuatnya.

> **Note**
>
> Kami telah mendesain ulang konsol EMR Amazon agar lebih mudah digunakan. Lihat Apa yang baru dengan konsol? untuk mempelajari tentang perbedaan antara pengalaman konsol lama dan baru.

New console

Untuk menambahkan tag saat Anda membuat klaster dengan konsol baru

1. Masuk keAWS Management Console, dan buka konsol EMR Amazon di https://console.aws.amazon.com/emr.
2. Di bawah EMR pada EC2 di panel navigasi kiri, pilih Clusters, lalu pilih Create cluster.
3. Di bawah Tag, pilih Tambahkan tag baru. Tentukan tag di bidang Kunci. Secara opsional, tentukan tag di bidang Nilai.
4. Pilih opsi lain yang berlaku untuk cluster Anda.
5. Untuk meluncurkan klaster Anda, pilih Buat klaster.

Old console

Untuk menambahkan tag saat Anda membuat cluster dengan konsol lama

1. Arahkan ke konsol EMR Amazon baru dan pilih Beralih ke konsol lama dari navigasi samping. Untuk informasi selengkapnya tentang apa yang diharapkan saat beralih ke konsol lama, lihat Menggunakan konsol lama.
2. Pilih Buat klaster, Pergi ke opsi lanjutan.
3. Pada halaman Langkah 3: Pengaturan Klaster Umum, di bagian Tag, ketik Kunci untuk tag Anda.

   Saat Anda mulai mengetik Kunci, sebuah baris baru akan secara otomatis muncul untuk memberi ruang bagi tag baru berikutnya.
4. Secar ppsional, ketik Nilai untuk tag.
5. Ulangi langkah sebelumnya untuk setiap pasangan kunci/nilai tag yang akan ditambahkan ke klaster. Ketika klaster diluncurkan, setiap tag yang Anda masukkan secara otomatis akan dikaitkan dengan klaster.

AWS CLI

Untuk menambahkan tag saat Anda membuat cluster dengan AWS CLI

Contoh berikut menunjukkan cara menambahkan tag ke klaster baru dengan menggunakan AWS CLI. Untuk menambahkan tag saat Anda membuat klaster, ketik subperintah `create-cluster` dengan parameter `--tags`.

- Untuk menambahkan tag bernama *`costCenter`* dengan *`pemasaran`* nilai kunci saat Anda membuat klaster, ketik perintah berikut dan ganti *`myKey`* dengan nama pasangan kunci EC2 Anda.

```
aws emr create-cluster --name "Test cluster" --release-label emr-4.0.0 --
applications Name=Hadoop Name=Hive Name=Pig --tags "costCenter=marketing" --
use-default-roles --ec2-attributes KeyName=myKey --instance-type m5.xlarge --
instance-count 3
```

Ketika Anda menentukan jumlah instans tanpa menggunakan `--instance-groups` parameter, Simpul utama tunggal diluncurkan, dan instans yang tersisa diluncurkan sebagai simpul inti. Semua simpul akan menggunakan tipe instans yang ditentukan dalam perintah.

> **Note**
>
> Jika Anda belum sebelumnya membuat peran layanan EMR default dan profil instans EC2, ketik aws `emr create-default-roles` untuk membuatnya sebelum mengetik `create-cluster` subperintah.

Untuk informasi selengkapnya tentang menggunakan perintah Amazon EMR dalam AWS CLI, lihat https://docs.aws.amazon.com/cli/latest/reference/emr.

Anda juga dapat menambahkan tag ke klaster yang sudah ada.

New console

Untuk menambahkan tag ke cluster yang ada dengan konsol baru

1. Masuk keAWS Management Console, dan buka konsol EMR Amazon di https://console.aws.amazon.com/emr.

2. Di bawah EMR pada EC2 di panel navigasi kiri, pilih Cluster, dan pilih cluster yang ingin Anda perbarui.
3. Pada tab Tag di halaman detail cluster, pilih Kelola tag. Tentukan tag di bidang Kunci. Secara opsional, tentukan tag di bidang Nilai.
4. Pilih Simpan perubahan. Tab Tag diperbarui dengan jumlah tag baru yang Anda miliki di cluster Anda. Misalnya, jika Anda sekarang memiliki dua tag, label tab Anda adalah Tag (2).

Old console

Untuk menambahkan tag ke cluster yang ada dengan konsol lama

1. Dalam konsol Amazon EMR, pilih Daftar klaster lalu klik klaster untuk menambahkan tag.
2. Pada halaman Rincian Klaster, di bidang Tag, klik Lihat Semua/Edit.
3. Pada halaman Lihat Semua/Edit, klik Tambahkan.
4. Klik bidang kosong di kolom Kunci lalu ketik nama kunci Anda.
5. Secara opsional, klik bidang kosong di kolom Nilai lalu ketik nama nilai Anda.
6. Dengan setiap tag baru yang Anda mulai, baris tag kosong lainnya akan muncul di bawah tag yang sedang Anda edit. Ulangi langkah-langkah sebelumnya pada baris tag baru untuk setiap tag yang akan ditambahkan.

AWS CLI

Untuk menambahkan tag ke cluster yang sedang berjalan dengan AWS CLI

- Masukkan `add-tags` subperintah dengan `--tag` parameter untuk menetapkan tag ke ID cluster. Anda dapat menemukan ID cluster menggunakan konsol atau `list-clusters` perintah. Subperintah `add-tags` saat ini hanya menerima satu ID sumber daya.

  Misalnya, untuk menambahkan dua tag ke klaster yang sedang berjalan dengan kunci bernama *`CostCenter`* dengan nilai *`pemasaran`* dan tag lain bernama *`lain`* dengan nilai *`akuntansi`*, masukkan perintah berikut dan ganti *`J-KT4xxxxxxxx1nm`* dengan ID cluster Anda.

  ```
  aws emr add-tags --resource-id j-KT4XXXXXXXX1NM --tag "costCenter=marketing" --
  tag "other=accounting"
  ```

Perhatikan bahwa ketika tag ditambahkan menggunakan AWS CLI, tidak ada output dari perintah. Untuk informasi selengkapnya tentang menggunakan perintah Amazon EMR dalam AWS CLI, lihat https://docs.aws.amazon.com/cli/latest/reference/emr.

Melihat tag pada klaster

Jika Anda ingin melihat semua tag yang terkait dengan cluster, Anda dapat melihatnya dengan konsol atauAWS CLI.

> **Note**
>
> Kami telah mendesain ulang konsol EMR Amazon agar lebih mudah digunakan. Lihat Apa yang baru dengan konsol? untuk mempelajari tentang perbedaan antara pengalaman konsol lama dan baru.

New console

Untuk melihat tag pada cluster dengan konsol baru

1. Masuk keAWS Management Console, dan buka konsol EMR Amazon di https://console.aws.amazon.com/emr.
2. Di bawah EMR pada EC2 di panel navigasi kiri, pilih Cluster, dan pilih cluster yang ingin Anda perbarui.
3. Untuk melihat semua tag Anda, pilih tab Tag pada halaman detail cluster.

Old console

Untuk melihat tag pada cluster dengan konsol lama

1. Dalam konsol Amazon EMR, pilih Daftar klaster lalu klik klaster untuk melihat tag.
2. Pada halaman Rincian klaster, di bidang Tag, beberapa tag ditampilkan di sini. Klik Lihat Semua/Edit untuk menampilkan semua tag yang tersedia di klaster.

AWS CLI

Untuk melihat tag pada cluster dengan AWS CLI

Untuk melihat tag pada sebuah klaster menggunakan AWS CLI, ketik subperintah `describe-cluster` dengan parameter `--query`.

- Untuk melihat tag klaster, ketik perintah berikut dan ganti *`j-KT4XXXXXXXX1NM`* dengan ID klaster Anda.

```
aws emr describe-cluster --cluster-id j-KT4XXXXXX1NM --query Cluster.Tags
```

Output ini menampilkan semua informasi tag tentang klaster yang mirip dengan yang berikut ini:

```
Value: accounting     Value: marketing
Key: other            Key: costCenter
```

Untuk informasi selengkapnya tentang menggunakan perintah Amazon EMR dalam AWS CLI, lihat https://docs.aws.amazon.com/cli/latest/reference/emr.

Menghapus tag dari sebuah klaster

Jika Anda tidak lagi membutuhkan tag, Anda dapat menghapusnya dari klaster.

> **Note**
>
> Kami telah mendesain ulang konsol EMR Amazon agar lebih mudah digunakan. Lihat Apa yang baru dengan konsol? untuk mempelajari tentang perbedaan antara pengalaman konsol lama dan baru.

New console

Untuk menghapus tag pada cluster dengan konsol baru

1. Masuk keAWS Management Console, dan buka konsol EMR Amazon di https://console.aws.amazon.com/emr.

2. Di bawah EMR pada EC2 di panel navigasi kiri, pilih Cluster, dan pilih cluster yang ingin Anda perbarui.
3. Pada tab Tag di halaman detail cluster, pilih Kelola tag.
4. Pilih Hapus untuk setiap pasangan kunci-nilai yang ingin Anda hapus.
5. Pilih Simpan perubahan.

Old console

Untuk menghapus tag pada cluster dengan konsol lama

1. Dalam konsol Amazon EMR, pilih halaman Daftar klaster lalu klik klaster untuk menghapus tag.
2. Pada halaman Rincian klaster, di bidang Tag, klik Lihat Semua/Edit.
3. Di kotak dialog Lihat Semua/Edit, klik ikon X di sebelah tag untuk menghapus lalu klik Simpan.
4. (Opsional) Ulangi langkah sebelumnya untuk setiap pasangan nilai kunci tag untuk dihapus dari cluster.

AWS CLI

Untuk menghapus tag pada cluster dengan AWS CLI

Ketik `remove-tags` subperintah dengan `--tag-keys` parameter. Saat menghapus tag, hanya nama kunci yang dibutuhkan.

- Untuk menghapus tag dari klaster, ketik perintah berikut ini dan ganti *`j-KT4XXXXXXXX1NM`* dengan ID klaster Anda.

```
aws emr remove-tags --resource-id j-KT4XXXXXX1NM --tag-keys "costCenter"
```

> **Note**
>
> Saat ini Anda tidak dapat menghapus beberapa tag menggunakan satu perintah.

Untuk informasi selengkapnya tentang menggunakan perintah Amazon EMR dalam AWS CLI, lihat https://docs.aws.amazon.com/cli/latest/reference/emr.

Driver dan integrasi aplikasi pihak ketiga

Anda dapat menjalankan beberapa aplikasi big data populer di Amazon EMR dengan harga utilitas. Ini berarti Anda membayar biaya nominal tambahan per jam untuk aplikasi pihak ketiga saat klaster Anda berjalan. Ini memungkinkan Anda untuk menggunakan aplikasi tanpa harus membeli lisensi tahunan. Bagian berikut menjelaskan beberapa alat yang dapat Anda gunakan dengan EMR.

Topik

Gunakan alat intelijen bisnis dengan Amazon EMR

Anda dapat menggunakan alat intelijen bisnis populer seperti Microsoft Excel,, MicroStrategyQlikView, dan Tableau dengan Amazon EMR untuk menjelajahi dan memvisualisasikan data Anda. Sebagian besar dari alat ini memerlukan driver ODBC (Open Database Connectivity) atau JDBC (Java Database Connectivity). Untuk mengunduh dan menginstal driver terbaru, lihathttp://awssupportdatasvcs.com/bootstrap-actions/Simba/latest/.

Untuk menemukan versi driver yang lebih lama, lihathttp://awssupportdatasvcs.com/bootstrap-actions/Simba/.

Keamanan di Amazon EMR

Keamanan cloud di AWS merupakan prioritas tertinggi. Sebagai seorang pelanggan AWS, Anda mendapatkan manfaat dari pusat data dan arsitektur jaringan yang dibangun untuk memenuhi persyaratan dari 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 ke tiga menguji dan memverifikasi efektivitas keamanan kami sebagai bagian dari AWS program kepatuhan secara berkala. Untuk mempelajari tentang program kepatuhan yang berlaku di Amazon EMR, lihat layanan AWS di cakupan melalui program kepatuhan.
- 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 korporasi Anda, serta undang-undang dan peraturan yang berlaku.

Dokumentasi ini membantu Anda memahami cara menerapkan model tanggung jawab bersama saat menggunakan Amazon EMR. Ketika Anda mengembangkan solusi di Amazon EMR, gunakan teknologi berikut untuk membantu mengamankan sumber daya klaster dan data sesuai dengan kebutuhan bisnis Anda. Topik berikut menunjukkan kepada Anda cara mengonfigurasi Amazon EMR dan menggunakan layanan AWS lain untuk memenuhi tujuan keamanan dan kepatuhan Anda.

Konfigurasi grup keamanan

Konfigurasi keamanan di Amazon EMR adalah templat untuk setup keamanan yang berbeda. Anda dapat membuat konfigurasi keamanan untuk mudah menggunakan kembali pengaturan keamanan setiap kali Anda membuat sebuah klaster. Untuk informasi selengkapnya, lihat Menggunakan konfigurasi keamanan untuk mengatur keamanan klaster.

Perlindungan data

Anda dapat menerapkan enkripsi data untuk membantu melindungi data at rest di Amazon S3, data at rest di penyimpanan instans klaster, dan data dalam transit. Untuk informasi selengkapnya, lihat Enkripsi data at rest dan dalam transit.

AWS Identity and Access Management dengan Amazon EMR

AWS Identity and Access Management (IAM) adalah layanan AWS yang membantu administrator mengontrol akses ke sumber daya AWS secara aman. Administrator IAM mengontrol siapa yang dapat diautentikasi (masuk) dan diotorisasi (memiliki izin) untuk menggunakan sumber daya Amazon EMR. IAM adalah layanan AWS yang dapat Anda gunakan tanpa biaya tambahan.

- Kebijakan Berbasis Identitas IAM — Kebijakan IAM mengizinkan atau menolak izin bagi pengguna dan grup untuk melakukan tindakan. Kebijakan dapat dikombinasikan dengan penandaan untuk mengontrol akses cluster-by-cluster berdasarkan. Untuk informasi selengkapnya, lihat AWS Identity and Access Management untuk Amazon EMR.
- IAM role – Peran layanan Amazon EMR, profil instans, dan kendali peran tertaut layanan bagaimana Amazon EMR dapat mengakses layanan AWS lain. Untuk informasi selengkapnya, lihat Konfigurasi peran layanan IAM untuk izin Amazon EMR untuk layanan AWS dan sumber daya.
- IAM role untuk permintaan EMRFS ke Amazon S3 – Ketika Amazon EMR mengakses Amazon S3, Anda dapat menentukan IAM role untuk menggunakan berdasarkan pengguna, grup, atau lokasi data EMRFS di Amazon S3. Hal ini mengizinkan Anda untuk secara tepat mengontrol apakah pengguna klaster dapat mengakses file dari di Amazon EMR. Untuk informasi selengkapnya, lihat Konfigurasi IAM role untuk permintaan EMRFS ke Amazon S3.

Kerberos

Anda dapat mengatur Kerberos untuk memberikan autentikasi yang kuat melalui kriptografi kunci rahasia. Untuk informasi selengkapnya, lihat Gunakan Kerberos untuk otentikasi dengan Amazon EMR.

Lake Formation

Anda dapat menggunakan izin Lake Formation bersama dengan AWS Katalog Data Glue untuk menyediakan akses level kolom ke basis data dan tabel di AWS Katalog Data Glue. Lake Formation mengaktifkan sistem masuk tunggal federasi ke EMR Notebooks atau Apache Zeppelin dari sistem identitas korporasi. Untuk informasi selengkapnya, lihat Integrasi Amazon EMR dengan AWS Lake Formation.

Secure Socket Shell (SSH)

SSH membantu memberikan cara yang aman bagi pengguna untuk connect ke baris perintah pada instans klaster. Hal ini juga menyediakan pembuatan terowongan untuk melihat antarmuka web yang aplikasi host pada simpul utama. Klien dapat mengautentikasi menggunakan Kerberos atau pasangan kunci Amazon EC2. Untuk informasi lebih lanjut, lihat Menggunakan key pair EC2 untuk kredensi SSH dan Connect ke sebuah cluster.

Grup keamanan Amazon EC2

Grup keamanan bertindak sebagai firewall virtual untuk instans klaster EMR, membatasi lalu lintas jaringan masuk dan keluar. Untuk informasi selengkapnya, lihat Mengendalikan lalu lintas jaringan dengan grup keamanan.

Default Amazon Linux AMI para Amazon EMR

> **Important**
>
> Cluster EMR yang menjalankan Amazon Linux atau Amazon Linux 2 Amazon Machine Images (AMI) menggunakan perilaku default Amazon Linux, dan tidak secara otomatis mengunduh dan menginstal pembaruan kernel penting dan kritis yang memerlukan reboot. Ini adalah perilaku yang sama dengan instans Amazon EC2 lainnya yang menjalankan AMI Amazon Linux default. Jika pembaruan perangkat lunak Amazon Linux baru yang memerlukan reboot (seperti pembaruan kernel, NVIDIA, dan CUDA) tersedia setelah rilis EMR Amazon tersedia, instance cluster EMR yang menjalankan AMI default tidak secara otomatis mengunduh dan menginstal pembaruan tersebut. Untuk mendapatkan pembaruan kernel, Anda dapat menyesuaikan Amazon EMR AMI menjadi gunakan Amazon Linux AMI terbaru.

Tergantung pada postur keamanan aplikasi Anda dan lama waktu berjalannya klaster, Anda dapat memilih untuk secara berkala me-reboot klaster Anda untuk menerapkan pembaruan keamanan, atau membuat tindakan bootstrap untuk menyesuaikan paket instalasi dan pembaruan. Anda juga dapat memilih untuk menguji dan versi terbaru menginstal memilih pembaruan keamanan pada menjalankan instans klaster. Untuk informasi selengkapnya, lihat Menggunakan AMI Amazon Linux default untuk Amazon EMR. Perhatikan bahwa konfigurasi jaringan Anda harus mengizinkan untuk HTTP dan HTTPS keluar ke repositori Amazon Linux di Amazon S3, jika tidak, pembaruan keamanan tidak akan berhasil.

Menggunakan konfigurasi keamanan untuk mengatur keamanan klaster

Anda dapat menggunakan konfigurasi keamanan Amazon EMR untuk mengonfigurasi enkripsi data, otentikasi Kerberos, dan otorisasi Amazon S3 untuk EMRFS di cluster Anda. Pertama, Anda membuat konfigurasi keamanan. Kemudian, konfigurasi keamanan tersedia untuk digunakan dan digunakan kembali saat Anda membuat cluster.

Anda dapat menggunakanAWS Management Console, AWS Command Line Interface (AWS CLI), atau AWS SDK untuk membuat konfigurasi keamanan. Anda juga dapat menggunakan templat AWS CloudFormation untuk membuat konfigurasi keamanan. Untuk informasi selengkapnya, lihat Panduan AWS CloudFormation Pengguna dan referensi templat untuk AWS::EMR::SecurityConfiguration.

Topik

- Membuat konfigurasi keamanan
- Menentukan konfigurasi keamanan untuk sebuah klaster

Membuat konfigurasi keamanan

Topik ini mencakup prosedur umum untuk membuat konfigurasi keamanan dengan konsol EMR Amazon danAWS CLI, diikuti dengan referensi untuk parameter yang terdiri dari enkripsi, otentikasi, dan peran IAM untuk EMRFS. Untuk informasi lebih lanjut tentang izin, lihat topik berikut:

- Enkripsi data at rest dan dalam transit
- Gunakan Kerberos untuk otentikasi dengan Amazon EMR
- Konfigurasi IAM role untuk permintaan EMRFS ke Amazon S3

Untuk membuat konfigurasi keamanan menggunakan konsol

1. Buka konsol EMR Amazon di https://console.aws.amazon.com/emr.
2. Di panel navigasi, memilih Konfigurasi Keamanan, Buat konfigurasi keamanan.
3. Ketik Nama untuk konfigurasi keamanan.
4. Memilih opsi untuk Enkripsi dan Autentikasi seperti yang dijelaskan pada bagian di bawah dan versi terbaru memilih Buat.

Untuk membuat konfigurasi keamanan menggunakan AWS CLI

- Gunakan perintah `create-security-configuration` seperti pada contoh berikut.
  - Untuk *`SecConfigName`*, tentukan nama konfigurasi keamanan. Ini adalah nama yang Anda tentukan saat Anda membuat sebuah klaster yang menggunakan konfigurasi keamanan ini.
  - Untuk *`SecConfigDef`*, tentukan struktur JSON inline atau jalur ke file JSON lokal, seperti *`file://MySecConfig.json`*. Parameter JSON menentukan pilihan untuk Enkripsi, IAM role untuk akses EMRFS ke Amazon S3, dan Autentikasi seperti yang dijelaskan pada bagian di bawah ini.

```
aws emr create-security-configuration --name "SecConfigName" --security-
configuration SecConfigDef
```

Konfigurasi enkripsi data

Sebelum Anda mengonfigurasi enkripsi di konfigurasi keamanan, buat kunci dan sertifikat yang digunakan untuk enkripsi. Untuk informasi lebih lanjut, lihat Menyediakan kunci untuk mengenkripsi data at rest dengan Amazon EMR dan Memberikan sertifikat untuk mengenkripsi data dalam transit dengan enkripsi Amazon EMR.

Bila Anda membuat konfigurasi keamanan, Anda menentukan dua rangkaian opsi enkripsi: enkripsi data yang tersisa dan enkripsi data dalam transit. Pilihan untuk enkripsi data yang tersisa mencakup Amazon S3 dengan EMRFS dan enkripsi disk lokal. Opsi enkripsi dalam transit mengaktifkan fitur enkripsi sumber terbuka untuk aplikasi tertentu yang mendukung Keamanan Lapisan Pengangkutan (TLS). Pilihan saat istirahat dan opsi dalam transit dapat diaktifkan secara bersamaan atau terpisah. Untuk informasi selengkapnya, lihat Enkripsi data at rest dan dalam transit.

> **Note**
>
> Saat Anda menggunakan AWS KMS, biaya berlaku untuk penyimpanan dan penggunaan kunci enkripsi. Untuk informasi lebih lanjut, lihat AWS KMS Harga.

Menentukan opsi enkripsi menggunakan konsol

Memilih opsi di bawah Enkripsi sesuai dengan panduan berikut.

- Memilih opsi di bawah Enkripsi saat diam untuk mengenkripsi data yang tersimpan di sistem file.

  Anda dapat memilih untuk mengenkripsi data di Amazon S3, disk lokal, atau keduanya.
- Di bawah Enkripsi data S3, untuk Mode enkripsi memilih nilai untuk menentukan bagaimana Amazon EMR mengenkripsi data Amazon S3 dengan EMRFS.

  Apa yang Anda lakukan selanjutnya tergantung pada mode enkripsi yang Anda pilih:

  - SSE-S3

    Tentukan Enkripsi sisi server dengan kunci enkripsi yang dikelola Amazon S3. Anda tidak perlu melakukan apa pun lagi karena Amazon S3 menangani kunci untuk Anda.
  - SSE-KMS atau CSE-KMS

    Menentukan enkripsi sisi server dengan AWS KMS kunci yang dikelola (SSE-KMS) atau enkripsi di sisi klien dengan AWS KMS kunci terkelola (CSE-KMS). Untuk AWS KMS key, pilih satu kunci. Kunci harus ada di wilayah yang sama dengan klaster EMR. Untuk persyaratan utama, lihat Menggunakan AWS KMS keys untuk enkripsi.
  - CSE-kustom

    Menentukan enkripsi sisi klien menggunakan kunci root sisi klien kustom (CSE-Custom). Untuk objek S3, masukkan lokasi di Amazon S3, atau Amazon S3 ARN, file JAR penyedia kunci kustom Anda. Kemudian, untuk kelas penyedia kunci, masukkan nama kelas lengkap dari kelas yang dideklarasikan dalam aplikasi Anda yang mengimplementasikan EncryptionMaterialsProvider antarmuka.
- Di bawah Enkripsi disk lokal, memilih nilai untuk Tipe penyedia kunci.
  - AWS KMS key

Pilih opsi ini untuk menentukan fileAWS KMS key. Untuk AWS KMS key, pilih satu kunci. Kunci harus ada di wilayah yang sama dengan klaster EMR. Untuk informasi lebih lanjut tentang kunci yang diperlukan, lihat Menggunakan AWS KMS keys untuk enkripsi.

Enkripsi EBS

Bila Anda menentukan AWS KMS sebagai penyedia kunci, Anda dapat mengaktifkan enkripsi EBS untuk mengenkripsi perangkat root EBS dan volume penyimpanan. Untuk mengaktifkan opsi tersebut, Anda harus memberikan peran layanan EMR Amazon `EMR_DefaultRole` dengan izin untuk menggunakan AWS KMS key yang Anda tentukan. Untuk informasi lebih lanjut tentang kunci yang diperlukan, lihat Mengaktifkan enkripsi EBS dengan memberikan izin tambahan untuk kunci KMS.

- Kustom

  Memilih opsi ini untuk menentukan penyedia kunci kustom. Untuk objek S3, masukkan lokasi di Amazon S3, atau Amazon S3 ARN, file JAR penyedia kunci kustom Anda. Untuk kelas penyedia Key, masukkan nama kelas lengkap dari kelas yang dideklarasikan dalam aplikasi Anda yang mengimplementasikan EncryptionMaterialsProvider antarmuka. Nama kelas yang Anda berikan di sini harus berbeda dari nama kelas yang disediakan untuk CSE-Custom.

- Memilih Enkripsi dalam transit untuk mengaktifkan fitur enkripsi TLS sumber terbuka untuk data dalam transit. Memilih Tipe penyedia sertifikat menurut panduan berikut:
  - PEM

    Memilih opsi ini untuk menggunakan file PEM yang Anda berikan di file zip. Dua artefak diperlukan di file zip: privateKey.pem dan certificateChain.pem. File ketiga, trustedCertificates.pem, adalah opsional. Lihat Memberikan sertifikat untuk mengenkripsi data dalam transit dengan enkripsi Amazon EMR untuk detail. Untuk objek S3, tentukan lokasi di Amazon S3, atau Amazon S3 ARN, bidang file zip.

  - Kustom

    Memilih opsi ini untuk menentukan penyedia sertifikat kustom dan versi terbaru, untuk objek S3, masukkan lokasi di Amazon S3, atau Amazon S3 ARN, file JAR penyedia sertifikat kustom Anda. Untuk kelas penyedia Key, masukkan nama kelas lengkap dari kelas yang dideklarasikan dalam aplikasi Anda yang mengimplementasikan antarmuka TLSArtifactsProvider .

Menentukan opsi enkripsi menggunakan AWS CLI

Bagian yang mengikuti skenario penggunaan sampel untuk mengcitrakan JSON --security-configuration yang dibentuk dengan baik untuk konfigurasi yang berbeda dan penyedia kunci, diikuti dengan referensi untuk parameter JSON dan nilai-nilai yang sesuai.

Contoh opsi enkripsi data dalam transit

Contoh berikut menggambarkan skenario berikut:

- Enkripsi data dalam transit diaktifkan dan enkripsi data yang tidak aktif dinonaktifkan.
- Sebuah file zip dengan sertifikat di Amazon S3 digunakan sebagai penyedia kunci (lihat Memberikan sertifikat untuk mengenkripsi data dalam transit dengan enkripsi Amazon EMR untuk persyaratan sertifikat.

```
aws emr create-security-configuration --name "MySecConfig" --security-configuration '{
 "EncryptionConfiguration": {
  "EnableInTransitEncryption": true,
  "EnableAtRestEncryption": false,
  "InTransitEncryptionConfiguration": {
   "TLSCertificateConfiguration": {
    "CertificateProviderType": "PEM",
    "S3Object": "s3://MyConfigStore/artifacts/MyCerts.zip"
   }
  }
 }
}'
```

Contoh berikut menggambarkan skenario berikut:

- Enkripsi data dalam transit diaktifkan dan enkripsi data yang tidak aktif dinonaktifkan.
- Sebuah penyedia kunci kustom digunakan (lihat Memberikan sertifikat untuk mengenkripsi data dalam transit dengan enkripsi Amazon EMR untuk persyaratan sertifikat).

```
aws emr create-security-configuration --name "MySecConfig" --security-configuration '{
 "EncryptionConfiguration": {
  "EnableInTransitEncryption": true,
```

```
  "EnableAtRestEncryption": false,
  "InTransitEncryptionConfiguration": {
   "TLSCertificateConfiguration": {
    "CertificateProviderType": "Custom",
    "S3Object": "s3://MyConfig/artifacts/MyCerts.jar",
    "CertificateProviderClass": "com.mycompany.MyCertProvider"
   }
  }
  }
}'
```

Contoh opsi enkripsi data yang tersisa

Contoh berikut menggambarkan skenario berikut:

- Enkripsi data dalam transit dinonaktifkan dan enkripsi data yang tidak aktif diaktifkan.
- SSE-S3 digunakan untuk enkripsi Amazon S3.
- Enkripsi disk lokal menggunakan AWS KMS sebagai penyedia kunci.

```
aws emr create-security-configuration --name "MySecConfig" --security-configuration '{
 "EncryptionConfiguration": {
  "EnableInTransitEncryption": false,
  "EnableAtRestEncryption": true,
  "AtRestEncryptionConfiguration": {
   "S3EncryptionConfiguration": {
    "EncryptionMode": "SSE-S3"
   },
   "LocalDiskEncryptionConfiguration": {
    "EncryptionKeyProviderType": "AwsKms",
    "AwsKmsKey": "arn:aws:kms:us-
east-1:123456789012:key/12345678-1234-1234-1234-123456789012"
   }
  }
  }
}'
```

Contoh berikut menggambarkan skenario berikut:

- Enkripsi data dalam transit diaktifkan dan referensi file zip dengan sertifikat PEM di Amazon S3, menggunakan ARN.
- SSE-KMS digunakan untuk enkripsi Amazon S3.

- Enkripsi disk lokal menggunakan AWS KMS sebagai penyedia kunci.

```
aws emr create-security-configuration --name "MySecConfig" --security-configuration '{
 "EncryptionConfiguration": {
  "EnableInTransitEncryption": true,
  "EnableAtRestEncryption": true,
  "InTransitEncryptionConfiguration": {
   "TLSCertificateConfiguration": {
    "CertificateProviderType": "PEM",
    "S3Object": "arn:aws:s3:::MyConfigStore/artifacts/MyCerts.zip"
   }
  },
  "AtRestEncryptionConfiguration": {
   "S3EncryptionConfiguration": {
    "EncryptionMode": "SSE-KMS",
    "AwsKmsKey": "arn:aws:kms:us-
east-1:123456789012:key/12345678-1234-1234-1234-123456789012"
   },
   "LocalDiskEncryptionConfiguration": {
    "EncryptionKeyProviderType": "AwsKms",
    "AwsKmsKey": "arn:aws:kms:us-
east-1:123456789012:key/12345678-1234-1234-1234-123456789012"
   }
  }
 }
}'
```

Contoh berikut menggambarkan skenario berikut:

- Enkripsi data dalam transit diaktifkan dan referensi file zip dengan sertifikat PEM di Amazon S3.
- CSE-KMS digunakan untuk enkripsi Amazon S3.
- Enkripsi disk lokal menggunakan penyedia kunci kustom yang direferensikan oleh ARN.

```
aws emr create-security-configuration --name "MySecConfig" --security-configuration '{
 "EncryptionConfiguration": {
  "EnableInTransitEncryption": true,
  "EnableAtRestEncryption": true,
  "InTransitEncryptionConfiguration": {
```

```
   "TLSCertificateConfiguration": {
    "CertificateProviderType": "PEM",
    "S3Object": "s3://MyConfigStore/artifacts/MyCerts.zip"
   }
  },
  "AtRestEncryptionConfiguration": {
   "S3EncryptionConfiguration": {
    "EncryptionMode": "CSE-KMS",
    "AwsKmsKey": "arn:aws:kms:us-
east-1:123456789012:key/12345678-1234-1234-1234-123456789012"
   },
   "LocalDiskEncryptionConfiguration": {
    "EncryptionKeyProviderType": "Custom",
    "S3Object": "arn:aws:s3:::artifacts/MyKeyProvider.jar",
    "EncryptionKeyProviderClass": "com.mycompany.MyKeyProvider"
   }
  }
 }
}'
```

Contoh berikut menggambarkan skenario berikut:

- Enkripsi data dalam transit diaktifkan dengan penyedia kunci kustom.
- CSE-Custom digunakan untuk data Amazon S3.
- Enkripsi disk lokal menggunakan penyedia kunci kustom.

```
aws emr create-security-configuration --name "MySecConfig" --security-configuration '{
 "EncryptionConfiguration": {
  "EnableInTransitEncryption": "true",
  "EnableAtRestEncryption": "true",
  "InTransitEncryptionConfiguration": {
   "TLSCertificateConfiguration": {
    "CertificateProviderType": "Custom",
    "S3Object": "s3://MyConfig/artifacts/MyCerts.jar",
    "CertificateProviderClass": "com.mycompany.MyCertProvider"
   }
  },
  "AtRestEncryptionConfiguration": {
   "S3EncryptionConfiguration": {
    "EncryptionMode": "CSE-Custom",
```

```
    "S3Object": "s3://MyConfig/artifacts/MyCerts.jar",
    "EncryptionKeyProviderClass": "com.mycompany.MyKeyProvider"
    },
   "LocalDiskEncryptionConfiguration": {
    "EncryptionKeyProviderType": "Custom",
    "S3Object": "s3://MyConfig/artifacts/MyCerts.jar",
    "EncryptionKeyProviderClass": "com.mycompany.MyKeyProvider"
   }
  }
 }
}'
```

Contoh berikut menggambarkan skenario berikut:

- Enkripsi data dalam transit dinonaktifkan dan enkripsi data yang tidak aktif diaktifkan.
- Enkripsi Amazon S3 diaktifkan dengan SSE-KMS.
- Beberapa AWS KMS kunci digunakan, satu per setiap bucket S3, dan pengecualian enkripsi diterapkan ke bucket S3 individual ini.
- Enkripsi disk lokal dinonaktifkan.

```
aws emr create-security-configuration --name "MySecConfig" --security-configuration '{
   "EncryptionConfiguration": {
     "AtRestEncryptionConfiguration": {
             "S3EncryptionConfiguration": {
           "EncryptionMode": "SSE-KMS",
           "AwsKmsKey": "arn:aws:kms:us-
east-1:123456789012:key/12345678-1234-1234-1234-123456789012",
           "Overrides": [
              {
                "BucketName": "sse-s3-bucket-name",
                "EncryptionMode": "SSE-S3"
              },
              {
                "BucketName": "cse-kms-bucket-name",
                "EncryptionMode": "CSE-KMS",
                "AwsKmsKey": "arn:aws:kms:us-
east-1:123456789012:key/12345678-1234-1234-1234-123456789012"
              },
              {
                "BucketName": "sse-kms-bucket-name",
                "EncryptionMode": "SSE-KMS",
```

```
                "AwsKmsKey": "arn:aws:kms:us-
east-1:123456789012:key/12345678-1234-1234-1234-123456789012"
              }
             ]
             }
           },
     "EnableInTransitEncryption": false,
      "EnableAtRestEncryption": true
  }
}'
```

Contoh berikut menggambarkan skenario berikut:

- Enkripsi data dalam transit dinonaktifkan dan enkripsi data yang tidak aktif diaktifkan.
- Enkripsi Amazon S3 diaktifkan dengan SSE-S3 dan enkripsi disk lokal dinonaktifkan.

```
aws emr create-security-configuration --name "MyS3EncryptionConfig" --security-
configuration '{
    "EncryptionConfiguration": {
        "EnableInTransitEncryption": false,
        "EnableAtRestEncryption": true,
        "AtRestEncryptionConfiguration": {
            "S3EncryptionConfiguration": {
                "EncryptionMode": "SSE-S3"
            }
        }
     }
}'
```

Contoh berikut menggambarkan skenario berikut:

- Enkripsi data dalam transit dinonaktifkan dan enkripsi data yang tidak aktif diaktifkan.
- Enkripsi disk lokal diaktifkan dengan AWS KMS sebagai penyedia kunci dan enkripsi Amazon S3 dinonaktifkan.

```
aws emr create-security-configuration --name "MyLocalDiskEncryptionConfig" --security-
configuration '{
    "EncryptionConfiguration": {
        "EnableInTransitEncryption": false,
        "EnableAtRestEncryption": true,
```

```
        "AtRestEncryptionConfiguration": {
            "LocalDiskEncryptionConfiguration": {
                "EncryptionKeyProviderType": "AwsKms",
                "AwsKmsKey": "arn:aws:kms:us-
east-1:123456789012:key/12345678-1234-1234-1234-123456789012"
            }
        }
    }
}'
```

Contoh berikut menggambarkan skenario berikut:

- Enkripsi data dalam transit dinonaktifkan dan enkripsi data yang tidak aktif diaktifkan.
- Enkripsi disk lokal diaktifkan dengan AWS KMS sebagai penyedia kunci dan enkripsi Amazon S3 dinonaktifkan.
- Enkripsi EBS diaktifkan.

```
aws emr create-security-configuration --name "MyLocalDiskEncryptionConfig" --security-
configuration '{
    "EncryptionConfiguration": {
        "EnableInTransitEncryption": false,
        "EnableAtRestEncryption": true,
        "AtRestEncryptionConfiguration": {
            "LocalDiskEncryptionConfiguration": {
                "EnableEbsEncryption": true,
                "EncryptionKeyProviderType": "AwsKms",
                "AwsKmsKey": "arn:aws:kms:us-
east-1:123456789012:key/12345678-1234-1234-1234-123456789012"
            }
        }
    }
}'
```

Referensi JSON untuk pengaturan enkripsi

Tabel berikut mencantumkan parameter JSON untuk pengaturan enkripsi dan memberikan Deskripsi nilai yang dapat diterima untuk setiap parameter.

Parameter	Deskripsi
"EnableInTransitEncryption" : betul | SALAH	Specify BETUL to enable in-transit encryption and SALAH to disable it. If omitted, salah is assumed, and in-transit encryption is disabled.
“EnableAtRestEncryption“: benar | SALAH	Specify BETUL to enable at-rest encryption and SALAH to disable it. If omitted, SALAH is assumed and at-rest encryption is disabled.
Parameter enkripsi dalam transit	
"InTransitEncryptionConfiguration" :	Specifies a collection of values used to configure in-transit encryption when EnableInTransitEncryption is betul.
“CertificateProviderType“: “PEM” | "Kustom"	Specifies whether to use PEM certificates referenced with a zipped file, or a Kustom certificate provider. If PEM is specified, S3Object must be a reference to the location in Amazon S3 of a zip file containing the certificates. If Custom is specified, S3Object must be a reference to the location in Amazon S3 of a JAR file, followed by a CertificateProviderClass entry.
“S3Object”: "" *ZipLocation* | "*JarLocation* "	Provides the location in Amazon S3 to a zip file when PEM is specified, or to a JAR file when Khusus is specified. The format can be a path (for example, s3:///artefak/ .zip MyConfig CertFiles) or an ARN (for example, arn:aws:s3: ::Kode/ .jar) MyCertProvider . If a zip file is specified, it must contain files named exactly privateKey.pem and certificateChain.pem . A file named trustedCertificates.pem is optional.

Parameter	Deskripsi
“CertificateProviderClass": "*MyClassID* ”	Required only if Khusus is specified for CertificateProviderType . *MyClassID* specifies a full class name declared in the JAR file, which implements the TLSArtifactsProvider interface. For example, com.mycompany. MyCertProvider .
Parameter enkripsi AT-rest	
"AtRestEncryptionConfiguration" :	Specifies a collection of values for at-rest encryption when EnableAtRestEncryption is betul, including Amazon S3 encryption and local disk encryption.
Parameter enkripsi Amazon S3	
“EncryptionConfigurationS3":	Specifies a collection of values used for Amazon S3 encryption with the Amazon EMR File System (EMRFS).
"EncryptionMode" : "SSE-S3" | "SSE-KMS" | "CSE-KMS" | "CSE-Custom"	Specifies the type of Amazon S3 encryption to use. If SSE-S3 is specified, no further Amazon S3 encryption values are required. If either SSE-KMS or CSE-KMS is specified, an AWS KMS key ARN must be specified as the AwsKmsKey value. If CSE-Custom is specified, S3Object and EncryptionKeyProviderClass values must be specified.
“AwsKmsKey": " *MyKeyARN*”	Required only when either SSE-KMS or CSE-KMS is specified for EncryptionMode . *MyKeyARN* must be a fully specified ARN to a key (for example, arn:aws:kms:us-east-1:123456789012:key/12345678-1234-1234-1234-123456789012).

Parameter	Deskripsi
"S3Object" : "*JarLocation* "	Required only when CSE-Custom is specified for CertificateProviderType . *JarLocation* provides the location in Amazon S3 to a JAR file. The format can be a path (for example, s3:///artefak/ .jar MyConfig MyKeyProvider) or an ARN (for example, arn:aws:s3: ::Kode/ .jar) MyKeyProvider .
“EncryptionKeyProviderClass": "ID *myS3 KeyClass*”	Required only when CSE-Custom is specified for EncryptionMode . *ID myS3 KeyClass* specifies a full class name of a class declared in the application that implements the EncryptionMaterialsProvider interface; for example, *com.mycompany.mys3 KeyProvider* .
Parameter enkripsi disk lokal	
"LocalDiskEncryptionConfiguration"	Specifies the key provider and corresponding values to be used for local disk encryption.
"EnableEbsEncryption": true | false	Specify betul to enable EBS encryption. EBS encryption encrypts the EBS root device volume and attached storage volumes. To use EBS encryption, you must specify AwsKms as your EncryptionKeyProviderType .
"EncryptionKeyProviderType": "AwsKms" | "Kustom"	Specifies the key provider. If AwsKms is specified, an KMS key ARN must be specified as the AwsKmsKey value. If **Khusus** is specified, S3Object and EncryptionKeyProviderClass values must be specified.

Parameter	Deskripsi
“AwsKmsKey :" *MyKeyARN*”	Required only when AwsKms is specified for Tipe. *MyKeyARN* must be a fully specified ARN to a key (for example, arn:aws:kms:us-east-1:123456789012:key/12345678-1234-1234-1234-456789012123).
"S3Object" : "*JarLocation* "	Required only when CSE-Custom is specified for CertificateProviderType . *JarLocation* provides the location in Amazon S3 to a JAR file. The format can be a path (for example, s3:///artefak/ .jar MyConfig MyKeyProvider) or an ARN (for example, arn:aws:s3: ::Kode/ .jar) MyKeyProvider .
"EncryptionKeyProviderClass" : "*MyLocalDiskKeyClassID* "	Required only when Khusus is specified for Tipe. *MyLocalDiskKeyClassID* specifies a full class name of a class declared in the application that implements the EncryptionMaterialsProvider interface; for example, *com.mycompany. MyLocalDiskKeyProvider* .

mengonfigurasi autentikasi Kerberos

Konfigurasi keamanan dengan pengaturan Kerberos hanya dapat digunakan oleh sebuah klaster yang dibuat dengan atribut Kerberos atau kesalahan terjadi. Untuk informasi selengkapnya, lihat Gunakan Kerberos untuk otentikasi dengan Amazon EMR. Kerberos hanya tersedia di Amazon EMR versi 5.10.0 dan yang lebih baru.

Menentukan pengaturan Kerberos menggunakan konsol

Memilih opsi di bawah Autentikasi Kerberos menurut panduan berikut.

Parameter		Deskripsi
Kerberos		Menentukan bahwa Kerberos diaktifkan untuk klaster yang menggunakan konfigurasi keamanan ini. Jika sebuah klaster menggunakan konfigurasi keamanan ini, klaster juga harus memiliki pengaturan Kerberos yang ditentukan atau terjadi kesalahan.
Penyedia	KDC khusus cluster	Menentukan bahwa Amazon EMR membuat KDC pada node utama dari setiap cluster yang menggunakan konfigurasi keamanan ini. Anda menentukan nama ranah dan kata sandi admin KDC ketika Anda membuat klaster. Anda dapat referensi KDC ini dari klaster lain, jika diperlukan. Membuat klaster tersebut menggunakan konfigurasi keamanan yang berbeda, menentukan KDC eksternal, dan menggunakan nama ranah dan kata sandi admin KDC yang Anda tentukan untuk KDC khusus klaster.
	KDC Eksternal	Hanya tersedia dengan Amazon EMR 5.20.0 dan yang lebih baru. Menentukan bahwa klaster menggunakan konfigurasi keamanan ini mengautentikasi utama Kerberos menggunakan server KDC di luar klaster. KDC tidak dibuat pada klaster. Ketika Anda membuat klaster, Anda menentukan nama ranah dan kata sandi admin KDC untuk KDC eksternal.
Tiket Seumur Hidup		Opsional. Menentukan periode tiket Kerberos mana yang valid yang dikeluarkan oleh KDC pada klaster yang menggunakan konfigurasi keamanan ini. Masa pakai tiket terbatas untuk alasan keamanan. Aplikasi klaster dan layanan perpanjangan tiket otomatis setelah mereka kedaluwarsa. Pengguna yang terhubung ke cluster melalui SSH menggunakan kredensyal Kerberos harus menjalankan `kinit` dari

Parameter		Deskripsi
		baris perintah node utama untuk memperbarui setelah tiket kedaluwarsa.
Kepercayaan lintas alam		Menentukan kepercayaan lintas ranah antara KDC khusus klaster pada klaster yang menggunakan konfigurasi keamanan ini dan KDC di ranah Kerberos yang berbeda. Utama (biasanya pengguna) dari ranah lain diautenti kasi ke klaster yang menggunakan konfigurasi ini. Konfigurasi tambahan di ranah Kerberos lainnya diperlukan. Untuk informasi selengkapnya, lihat Tutorial: Konfigurasi kepercayaan lintas ranah dengan domain Direktori Aktif.
Properti kepercaya an lintas ranah	Realm	Menentukan nama ranah Kerberos dari ranah lain di hubungan kepercayaan. Dengan konvensi, nama ranah Kerberos adalah sama dengan nama domain tetapi semuanya menggunakan huruf kapital.
	Domain	Menentukan nama domain dari ranah lain di hubungan kepercayaan.
	Server admin	Menentukan nama domain yang memenuhi syarat (FQDN) atau alamat IP dari server admin di ranah lain dari hubungan kepercayaan. server admin dan server KDC biasanya berjalan pada mesin yang sama dengan FQDN yang sama, tetapi berkomunikasi pada port yang berbeda. Jika port tidak ditentukan, port 749 digunakan, yang merupakan default Kerberos. Atau, Anda dapat menentukan port (misalnya, `domain.example.com:749`).

Parameter		Deskripsi
	Server KDC	Menentukan nama domain yang memenuhi syarat (FQDN) atau alamat IP server KDC di ranah lain dari hubungan kepercayaan. Server KDC dan server admin biasanya berjalan pada mesin yang sama dengan FQDN yang sama, tetapi menggunakan port yang berbeda. Jika port tidak ditentukan, port 88 digunakan, yang merupakan default Kerberos. Atau, Anda dapat menentukan port (misalnya, `domain.example.com:88`).
KDC Eksternal		Menentukan bahwa klaster eksternal KDC digunakan oleh klaster.
Properti KDC eksternal	Server admin	Menentukan nama domain yang memenuhi syarat (FQDN) atau alamat IP dari server admin eksternal. Server admin dan server KDC biasanya berjalan pada mesin yang sama dengan FQDN yang sama, tetapi berkomunikasi pada port yang berbeda. Jika port tidak ditentukan, port 749 digunakan, yang merupakan default Kerberos. Atau, Anda dapat menentukan port (misalnya, `domain.example.com:749`).
	Server KDC	Menentukan nama domain yang memenuhi syarat (FQDN) dari server KDC eksternal. Server KDC dan server admin biasanya berjalan pada mesin yang sama dengan FQDN yang sama, tetapi menggunakan port yang berbeda. Jika port tidak ditentukan, port 88 digunakan, yang merupakan default Kerberos. Atau, Anda dapat menentukan port (misalnya, `domain.example.com:88`).

Parameter			Deskripsi
	Integrasi Direktori Aktif		Menentukan bahwa autentikasi utama Kerberos terintegrasi dengan domain Direktori Aktif Microsoft.
	Properti integrasi Direktori Aktif	Ranah Direktori Aktif	Menentukan nama ranah Kerberos dari domain Direktori Aktif. Dengan konvensi, nama ranah Kerberos biasanya sama dengan nama domain tetapi di huruf kapital semua.
		Domain Direktori Aktif	Menentukan nama domain Direktori Aktif.
		Server Direktori Aktif	Menentukan nama domain yang memenuhi syarat (FQDN) dari pengendali domain Direktori Aktif Microsoft.

Menentukan pengaturan Kerberos menggunakan AWS CLI

Tabel referensi berikut menunjukkan parameter JSON untuk pengaturan Kerberos di konfigurasi keamanan. Contoh konfigurasi, lihat, Contoh konfigurasi.

Parameter			Deskripsi
`"AuthenticationConfiguration": {`			Diperlukan untuk Kerberos. Menentukan bahwa konfigurasi autentikasi adalah bagian dari konfigurasi keamanan ini.
	`"KerberosConfiguration": {`		Diperlukan untuk Kerberos. Menentukan properti konfigurasi Kerberos.
		`"Provider": "ClusterDedicatedKdc",` —atau—	`ClusterDedicatedKdc` menetapkan bahwa Amazon EMR membuat KDC pada node utama dari setiap cluster yang

Parameter			Deskripsi
		"Provider: *"ExternalKdc",*	menggunakan konfigurasi keamanan ini. Anda menentukan nama ranah dan kata sandi admin KDC ketika Anda membuat klaster. Anda dapat referensi KDC ini dari klaster lain, jika diperlukan. Membuat klaster tersebut menggunakan konfigurasi keamanan yang berbeda, menentuka n KDC eksternal, dan menggunakan nama ranah dan kata sandi admin KDC yang Anda tentukan ketika Anda membuat klaster dengan KDC khusus klaster. *ExternalKdc* menentukan bahwa klaster menggunakan KDC eksternal . Amazon EMR tidak membuat KDC pada node utama. Klaster yang menggunakan konfigurasi keamanan ini harus menentukan nama ranah dan kata sandi admin KDC eksternal KDC.
		"ClusterDedicatedK dcConfiguration": {	Diperlukan ketika *ClusterDe dicatedKdc* ditentukan.

Parameter	Deskripsi
`"TicketLifetimeInHours": 24,`	Opsional. Menentukan periode tiket Kerberos mana yang valid yang dikeluarkan oleh KDC pada klaster yang menggunakan konfigurasi keamanan ini. Masa pakai tiket terbatas untuk alasan keamanan. Aplikasi klaster dan layanan perpanjangan tiket otomatis setelah mereka kedaluwarsa. Pengguna yang terhubung ke cluster melalui SSH menggunakan kredensyal Kerberos harus menjalankan `kinit` dari baris perintah node utama untuk memperbarui setelah tiket kedaluwarsa.
`"CrossRealmTrustConfiguration": {`	Menentukan kepercayaan lintas ranah antara KDC khusus klaster pada klaster yang menggunakan konfigurasi keamanan ini dan KDC di ranah Kerberos yang berbeda. Utama (biasanya pengguna) dari ranah lain diautentikasi ke klaster yang menggunakan konfigurasi ini. Konfigurasi tambahan di ranah Kerberos lainnya diperlukan. Untuk informasi selengkapnya, lihat Tutorial: Konfigurasi kepercayaan lintas ranah dengan domain Direktori Aktif.

Parameter	Deskripsi
"Realm": "*KDC2.COM*",	Menentukan nama ranah Kerberos dari ranah lain di hubungan kepercayaan. Dengan konvensi, nama ranah Kerberos adalah sama dengan nama domain tetapi semuanya menggunakan huruf kapital.
"Domain": "*kdc2.com*",	Menentukan nama domain dari ranah lain di hubungan kepercayaan.
"AdminServer": "*kdc.com:749* ",	Menentukan nama domain yang memenuhi syarat (FQDN) atau alamat IP dari server admin di ranah lain dari hubungan kepercayaan. server admin dan server KDC biasanya berjalan pada mesin yang sama dengan FQDN yang sama, tetapi berkomunikasi pada port yang berbeda. Jika port tidak ditentukan, port 749 digunakan, yang merupakan default Kerberos. Atau, Anda dapat menentukan port (misalnya, domain.example.com *:749*).

Parameter	Deskripsi
"KdcServer": "*kdc.com:88* "	Menentukan nama domain yang memenuhi syarat (FQDN) atau alamat IP server KDC di ranah lain dari hubungan kepercayaan. Server KDC dan server admin biasanya berjalan pada mesin yang sama dengan FQDN yang sama, tetapi menggunakan port yang berbeda. Jika port tidak ditentukan, port 88 digunakan, yang merupakan default Kerberos. Atau, Anda dapat menentukan port (misalnya, `domain.example.com` *:88*).
}	
}	
"ExternalKdcConfiguration": {	Diperlukan ketika *ExternalKdc* ditentukan.

Parameter				Deskripsi
			`"TicketLifetimeInHours": 24,`	Opsional. Menentukan periode tiket Kerberos mana yang valid yang dikeluarkan oleh KDC pada klaster yang menggunakan konfigurasi keamanan ini. Masa pakai tiket terbatas untuk alasan keamanan. Aplikasi klaster dan layanan perpanjangan tiket otomatis setelah mereka kedaluwarsa. Pengguna yang terhubung ke cluster melalui SSH menggunakan kredensyal Kerberos harus menjalankan `kinit` dari baris perintah node utama untuk memperbarui setelah tiket kedaluwarsa.
			`"KdcServerType": "Single",`	Menentukan bahwa satu server KDC direferensikan. `Single` saat ini adalah satu-satunya nilai yang didukung.

Parameter	Deskripsi
“AdminServer“:" *kdc.com:749* “,	Menentukan nama domain yang memenuhi syarat (FQDN) atau alamat IP dari server admin eksternal . Server admin dan server KDC biasanya berjalan pada mesin yang sama dengan FQDN yang sama, tetapi berkomunikasi pada port yang berbeda. Jika port tidak ditentukan, port 749 digunakan, yang merupakan default Kerberos. Atau, Anda dapat menentukan port (misalnya, `domain.example.com` *:749*).
“KdcServer“:" *kdc.com:88* “,	Menentukan nama domain yang memenuhi syarat (FQDN) dari server KDC eksternal. Server KDC dan server admin biasanya berjalan pada mesin yang sama dengan FQDN yang sama, tetapi menggunakan port yang berbeda. Jika port tidak ditentukan, port 88 digunakan, yang merupakan default Kerberos. Atau, Anda dapat menentukan port (misalnya, `domain.example.com` *:88*).
`"AdIntegrationConfiguration": {`	Menentukan bahwa autentikasi utama Kerberos terintegrasi dengan domain Direktori Aktif Microsoft.

Parameter	Deskripsi
"AdRealm": "*AD.DOMAIN.COM* ",	Menentukan nama ranah Kerberos dari domain Direktori Aktif. Dengan konvensi, nama ranah Kerberos biasanya sama dengan nama domain tetapi di huruf kapital semua.
"AdDomain": "*ad.domain.com* "	Menentukan nama domain Direktori Aktif.
"AdServer": "*ad.domain.com* "	Menentukan nama domain yang memenuhi syarat (FQDN) dari pengendali domain Direktori Aktif Microsoft.
}	
}	
}	
}	

Konfigurasi IAM role untuk permintaan EMRFS ke Amazon S3

IAM role untuk EMRFS mengizinkan Anda untuk memberikan izin yang berbeda untuk data EMRFS di Amazon S3. Anda membuat pemetaan yang menentukan IAM role yang digunakan untuk izin ketika permintaan akses berisi pengidentifikasi yang Anda tentukan. Pengidentifikasi dapat menjadi pengguna atau peran Hadoop, atau prefiks Amazon S3.

Untuk informasi selengkapnya, lihat Konfigurasi IAM role untuk permintaan EMRFS ke Amazon S3.

Menentukan IAM role untuk EMRFS menggunakan AWS CLI

Berikut ini adalah contoh potongan JSON untuk menentukan IAM role kustom untuk EMRFS di konfigurasi keamanan. Ini menunjukkan pemetaan peran untuk tiga tipe pengidentifikasi yang berbeda, diikuti dengan referensi parameter.

```
{
  "AuthorizationConfiguration": {
    "EmrFsConfiguration": {
      "RoleMappings": [{
        "Role": "arn:aws:iam::123456789101:role/allow_EMRFS_access_for_user1",
        "IdentifierType": "User",
        "Identifiers": [ "user1" ]
      },{
        "Role": "arn:aws:iam::123456789101:role/allow_EMRFS_access_to_MyBuckets",
        "IdentifierType": "Prefix",
        "Identifiers": [ "s3://MyBucket/","s3://MyOtherBucket/" ]
      },{
        "Role": "arn:aws:iam::123456789101:role/allow_EMRFS_access_for_AdminGroup",
        "IdentifierType": "Group",
        "Identifiers": [ "AdminGroup" ]
      }]
    }
  }
}
```

Parameter	Deskripsi
`"AuthorizationConfiguration":`	Diperlukan.
`"EmrFsConfiguration":`	Diperlukan. Berisi pemetaan peran.
`"RoleMappings":`	Diperlukan. Berisi satu atau lebih definisi peran pemetaan. Pemetaan peran dievaluasi di urutan top-down yang muncul. Jika pemetaan peran mengevaluasi sebagai BETUL untuk panggilan EMRFS untuk data di Amazon S3, tidak ada pemetaan peran lebih lanjut dievaluasi dan EMRFS menggunakan IAM role yang ditentukan untuk permintaan. Pemetaan peran terdiri dari parameter wajib berikut:
`"Role":`	Menentukan pengidentifikasi ARN dari IAM role dalam format `arn:aws:iam::` *account-id* `:role/`*role-name*. Ini adalah IAM role yang Amazon EMR asumsikan jika permintaa

Parameter	Deskripsi
	n EMRFS ke Amazon S3 cocok dengan salah satu `Identifiers` yang ditentukan.
`"IdentifierType":`	Dapat menjadi salah satu dari yang berikut: • `"User"` menetapkan bahwa pengidentifikasi adalah satu pengguna Hadoop atau lebih, yang bisa saja pengguna akun Linux atau utama Kerberos. Ketika permintaan EMRFS berasal dari pengguna atau pengguna yang ditentukan, IAM role diasumsikan. • `"Prefix"` menetapkan bahwa pengident ifikasi adalah lokasi Amazon S3. IAM role diasumsikan untuk panggilan ke lokasi atau lokasi dengan prefiks tertentu. Misalnya, prefiks `s3://mybucket/` mencocokkan `s3://mybucket/mydir` dan `s3://mybucket/yetanotherdir` . • `"Group"` menetapkan bahwa pengident ifikasi adalah satu Grup Hadoop atau lebih. IAM role diasumsikan jika permintaan berasal dari pengguna di grup atau grup-grup tertentu.
`"Identifiers":`	Menentukan satu pengidentifikasi atau lebih dari tipe pengidentifikasi yang sesuai. Pisahkan beberapa pengidentifikasi dengan koma tanpa spasi.

Konfigurasi permintaan layanan metadata untuk instans Amazon EC2

Metadata instans adalah data tentang instans Anda yang dapat Anda gunakan untuk mengonfigurasi atau mengelola instans yang sedang berjalan. Anda dapat mengakses metadata instans dari instans yang sedang berjalan menggunakan salah satu metode berikut:

- Layanan Metadata Instans Versi 1 (IMDSv1) - metode permintaan/tanggapan
- Layanan Metadata Instans Versi 2 (IMDSv2) - metode berorientasi sesi

Sementara Amazon EC2 mendukung IMDSv1 dan IMDSv2, Amazon EMR mendukung IMDSv2 di Amazon EMR 5.23.1, 5.27.1, 5.32 atau yang lebih baru, dan 6.2 atau yang lebih baru. di rilis ini, Amazon EMR komponen menggunakan IMDSv2 untuk semua panggilan IMDS. Untuk panggilan IMDS di kode aplikasi Anda, Anda dapat menggunakan IMDSv1 dan IMDSv2, atau mengonfigurasi IMDS untuk menggunakan hanya IMDSv2 untuk keamanan tambahan. Saat Anda menentukan IMDSv2 harus digunakan, maka IMDSv1 tidak lagi berfungsi.

Untuk informasi selengkapnya, lihat Mengonfigurasi layanan metadata instans di Panduan Pengguna Amazon EC2 untuk Instans Linux.

> **Note**
>
> Di Amazon EMR rilis 5.x atau 6.x sebelumnya, mematikan IMDSv1 menyebabkan kegagalan startup klaster sebagai Amazon EMR komponen menggunakan IMDSv1 untuk semua panggilan IMDS. Ketika mematikan IMDSv1, pastikan bahwa setiap perangkat lunak kustom yang menggunakan IMDSv1 diperbarui untuk IMDSv2.

Menentukan konfigurasi layanan metadata instans menggunakan AWS CLI

Berikut ini adalah contoh potongan JSON untuk menentukan instans Amazon EC2 metadata service (IMDS) di konfigurasi keamanan.

```
{
  "InstanceMetadataServiceConfiguration" : {
      "MinimumInstanceMetadataServiceVersion": integer,
      "HttpPutResponseHopLimit": integer
   }
}
```

Parameter	Deskripsi
`"InstanceMetadataServiceConfiguration":`	Diperlukan.

Parameter	Deskripsi
`"MinimumInstanceMetadataServiceVersion":`	Diperlukan. Tentukan 1 atau 2. Nilai dari 1 mengizinkan IMDSv1 dan IMDSv2. Nilai dari 2 hanya mengizinkan IMDSv2.
`"HttpPutResponseHopLimit":`	Diperlukan. Batas respons hop HTTP PUT yang diinginkan untuk permintaan metadata instans. Semakin besar jumlahnya, permintaan metadata instans lebih lanjut dapat melakukan perjalanan. Default: 1. Tentukan integer dari 1 ke 64.

Menentukan konfigurasi layanan metadata instans menggunakan konsol

Anda dapat mengonfigurasi penggunaan IMDS untuk sebuah klaster ketika Anda meluncurkannya dari konsol Amazon EMR.
Kendali konfigurasi Keamanan IMDS di konsol Amazon EMR

Untuk mengonfigurasi penggunaan IMDS menggunakan konsol:

1. Saat membuat konfigurasi keamanan baru di halaman Konfigurasi keamananmemilih Konfigurasi layanan metadata Instans EC2 di bawah Pengaturan Layanan Metadata Instans EC2. Konfigurasi ini hanya didukung di Amazon EMR 5.23.1, 5.27.1, 5.32 atau yang lebih baru, dan 6.2 atau yang lebih baru.
2. Untuk opsi Layanan Metadata Instans Versi memilih salah satu:
   - Matikan IMDSv1 dan hanya mengizinkan IMDSv2, jika Anda ingin mengizinkan hanya IMDSv2 pada klaster ini. Lihat Transisi ke layanan metadata instans versi 2 di Panduan Pengguna Amazon EC2 untuk Instans Linux.
   - Mengizinkan IMDSv1 dan IMDSv2 di klaster, jika Anda ingin mengizinkan IMDSv1 dan IMDSv2 orientasi sesi pada klaster ini.
3. Untuk IMDSv2, Anda juga dapat mengonfigurasi jumlah lompatan jaringan yang diizinkan untuk token metadata dengan pengaturan HTTP menempatkan respon hop batas untuk integer antara 1 dan 64.

Untuk informasi selengkapnya, lihat Mengonfigurasi layanan metadata instans di Panduan Pengguna Amazon EC2 untuk Instans Linux.

Lihat Konfigurasi detail instans dan Konfigurasi layanan metadata instans di Panduan Pengguna Amazon EC2 untuk Instans Linux.

Menentukan konfigurasi keamanan untuk sebuah klaster

Anda dapat menentukan pengaturan enkripsi ketika Anda membuat sebuah klaster dengan menentukan konfigurasi keamanan. Anda dapat menggunakan AWS Management Console atau AWS CLI.

> **Note**
>
> Kami telah mendesain ulang konsol EMR Amazon agar lebih mudah digunakan. Lihat Apa yang baru dengan konsol? untuk mempelajari tentang perbedaan antara pengalaman konsol lama dan baru.

New console

Untuk menentukan konfigurasi keamanan dengan konsol baru

1. Masuk keAWS Management Console, dan buka konsol EMR Amazon di https://console.aws.amazon.com/emr.
2. Di bawah EMR pada EC2 di panel navigasi kiri, pilih Clusters, lalu pilih Create cluster.
3. Di bawah Konfigurasi dan izin keamanan, temukan bidang Konfigurasi keamanan. Pilih menu tarik-turun atau pilih Browse untuk memilih nama konfigurasi keamanan yang Anda buat sebelumnya. Atau, pilih Buat konfigurasi keamanan untuk membuat konfigurasi yang dapat Anda gunakan untuk klaster Anda.
4. Pilih opsi lain yang berlaku untuk cluster Anda.
5. Untuk meluncurkan klaster Anda, pilih Buat klaster.

Old console

Untuk menentukan konfigurasi keamanan dengan konsol lama

1. Buka konsol EMR Amazon di https://console.aws.amazon.com/emr.

2. Pilih Buat klaster, Buka opsi lanjutan.
3. Pada Langkah 1: Perangkat Lunak dan Langkah layar, dari daftar Rilis, pilih emr-4.8.0 atau rilis yang lebih baru. Memilih pengaturan yang Anda inginkan dan memilih Selanjutnya.
4. Pada Langkah 2: Perangkat keras memilih pengaturan yang Anda inginkan dan memilih Selanjutnya. Lakukan hal yang sama untuk Langkah 3: Pengaturan Klaster Umum.
5. Pada layar Langkah 4: Keamanan, di bawah Opsi Enkripsi memilih nilai untuk Konfigurasi keamanan.
6. Konfigurasikan opsi keamanan lain seperti yang diinginkan dan memilih Buat klaster.

CLI

Untuk menentukan konfigurasi keamanan dengan AWS CLI

- Gunakan `aws emr create-cluster` untuk secara opsional menerapkan konfigurasi keamanan dengan`--security-configuration` *`MySecConfig`*, di mana *`MySecConfig`* adalah nama konfigurasi keamanan, seperti yang ditunjukkan pada contoh berikut. Yang `--release-label` Anda tentukan harus 4.8.0 atau yang lebih baru dan `--instance-type` dapat tersedia.

```
aws emr create-cluster --instance-type m5.xlarge --release-label emr-5.0.0 --
security-configuration mySecConfig
```

Perlindungan Data di Amazon EMR

AWS model tanggung jawab bersama diterapkan untuk perlindungan data di Amazon EMR. Sebagaimana dijelaskan di model ini, AWSbertanggung jawab untuk melindungi infrastruktur global yang menjalankan semua AWS Cloud. Anda bertanggung jawab untuk menjaga kendali terhadap konten Anda yang dihosting pada infrastruktur ini. Konten ini meliputi konfigurasi keamanan dan tugas pengelolaan untuk berbagai layanan AWS yang Anda gunakan. Untuk informasi selengkapnya tentang privasi data, lihat Pertanyaan Umum Privasi Data . Untuk informasi tentang perlindungan data di Eropa, lihat postingan blog model tanggung jawab bersama dan Peraturan Perlindungan Data Umum (GDPR) di AWS Blog Keamanan.

Untuk tujuan perlindungan data, kami menyarankan Anda untuk melindungi kredensyal AWS akun dan mengatur akun individual dengan. AWS Identity and Access Management Dengan cara seperti itu, setiap pengguna hanya diberi izin yang diperlukan untuk memenuhi tanggung jawab tugas

mereka. Kami juga merekomendasikan agar Anda mengamankan data Anda dengan cara-cara berikut ini:

- Gunakan otentikasi multifaktor (MFA) pada setiap akun.
- Menggunakan TLS untuk berkomunikasi dengan sumber daya AWS. Kami membutuhkan TLS 1.2.
- Menyiapkan API dan log aktivitas pengguna dengan AWS CloudTrail.
- Gunakan solusi enkripsi AWS, bersama dengan semua kontrol keamanan default dalam layanan AWS.
- Gunakan layanan keamanan terkelola lanjutan seperti Amazon Macie, yang membantu menemukan dan mengamankan data pribadi yang disimpan di Amazon Simple Storage Service (Amazon S3).
- Jika Anda memerlukan modul kriptografi tervalidasi FIPS 140-2 ketika mengakses AWS melalui antarmuka baris perintah atau API, gunakan titik akhir FIPS. Untuk informasi lebih lanjut tentang titik akhir FIPS yang tersedia, lihat Standar Pemrosesan Informasi Federal (FIPS) 140-2.

Kami sangat merekomendasikan agar Anda tidak memasukkan informasi identifikasi sensitif apapun, seperti nomor rekening pelanggan Anda, ke dalam kolom isian teks bebas seperti kolom Nama. Hal ini termasuk ketika Anda bekerja dengan Amazon EMR atau layanan AWS lainnya dengan menggunakan konsol, API, AWS CLI, atau SDK AWS. Data apa pun yang Anda masukkan ke dalam Amazon EMR atau layanan lain mungkin akan diambil untuk dimasukkan ke dalam log diagnostik. Saat Anda memberikan URL ke server eksternal, jangan menyertakan informasi kredensial di URL untuk memvalidasi permintaan Anda ke server tersebut.

Enkripsi data at rest dan dalam transit

Enkripsi data membantu mencegah pengguna yang tidak sah membaca data pada klaster dan sistem penyimpanan data terkait. Ini termasuk data yang disimpan ke media persisten, yang dikenal sebagai data at rest, dan data yang mungkin dicegat saat perjalanan jaringan, yang dikenal sebagai data dalam transit.

Dimulai dengan Amazon EMR versi 4.8.0, Anda dapat menggunakan konfigurasi keamanan Amazon EMR untuk mengonfigurasi pengaturan enkripsi data untuk klaster lebih mudah. Konfigurasi keamanan menawarkan pengaturan untuk mengaktifkan keamanan untuk data dalam transit dan data at rest di volume Amazon Elastic Block Store (Amazon EBS) dan EMRFS di Amazon S3.

Opsional, dimulai dengan Amazon EMR rilis versi 4.1.0 dan versi terbaru, Anda dapat memilih untuk mengonfigurasi enkripsi transparan di HDFS, yang tidak dikonfigurasi menggunakan konfigurasi

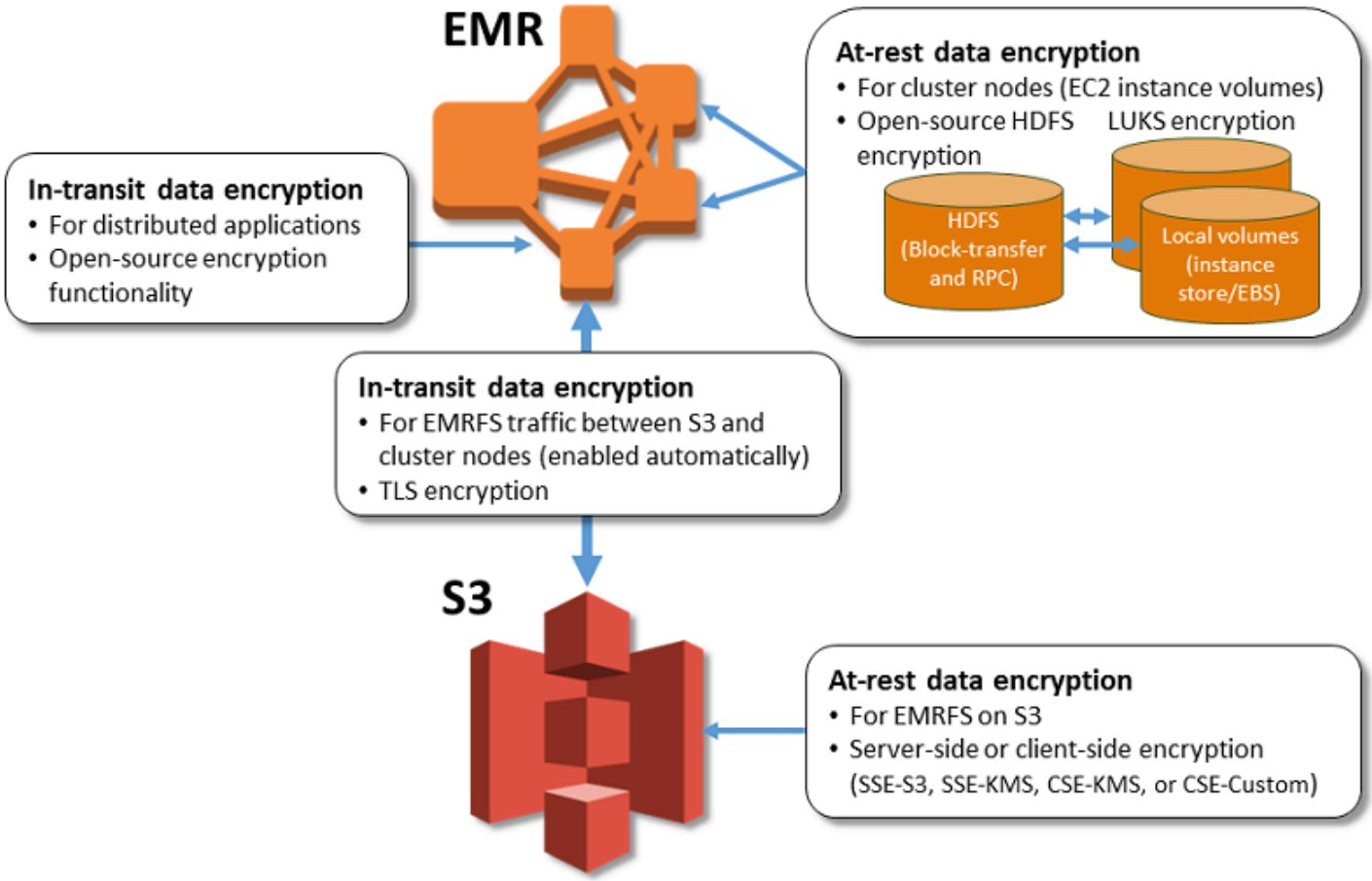
keamanan. Untuk informasi selengkapnya, lihat Enkripsi transparan di HDFS di Amazon EMR di Panduan Amazon EMR rilis.

Topik

Opsi enkripsi

Dengan Amazon EMR versi 4.8.0 dan yang lebih baru, Anda dapat menggunakan konfigurasi keamanan untuk menentukan pengaturan untuk mengenkripsi data at rest, data dalam transit, atau keduanya. Bila Anda mengaktifkan enkripsi data yang tersisa, Anda dapat memilih untuk mengenkripsi data EMRFS di Amazon S3, data di disk lokal, atau keduanya. Setiap konfigurasi keamanan yang Anda buat disimpan di Amazon EMR daripada di konfigurasi klaster, sehingga Anda dapat dengan mudah menggunakan kembali konfigurasi untuk menentukan pengaturan enkripsi data setiap kali Anda membuat sebuah klaster. Untuk informasi selengkapnya, lihat Membuat konfigurasi keamanan.

Diagram berikut menunjukkan pilihan enkripsi data yang berbeda tersedia dengan konfigurasi keamanan.



Opsi enkripsi berikut juga tersedia dan tidak dikonfigurasi menggunakan konfigurasi keamanan:

- Opsional, dengan Amazon EMR versi 4.1.0 dan versi terbaru, Anda dapat memilih untuk mengonfigurasi enkripsi transparan di HDFS. Untuk informasi selengkapnya, lihat Enkripsi transparan di HDFS di Amazon EMR di Panduan Amazon EMR rilis.
- Jika Anda menggunakan versi Amazon EMR rilis yang tidak mendukung konfigurasi keamanan, Anda dapat mengonfigurasi enkripsi untuk data EMRFS di Amazon S3 secara manual. Untuk informasi selengkapnya, lihat Menentukan enkripsi Amazon S3 menggunakan properti EMRFS.
- Jika Anda menggunakan versi Amazon EMR lebih awal dari 5.24.0, volume perangkat asal EBS yang dienkripsi didukung hanya bila menggunakan AMI kustom. Untuk informasi selengkapnya, lihat Membuat AMI kustom dengan volume perangkat root Amazon EBS terenkripsi di Panduan Manajemen EMR Amazon.

> **Note**
>
> Dimulai dengan Amazon EMR versi 5.24.0, Anda dapat menggunakan opsi konfigurasi keamanan untuk mengenkripsi perangkat asal EBS dan volume penyimpanan ketika Anda menentukan AWS KMS sebagai penyedia kunci Anda. Untuk informasi selengkapnya, lihat Enkripsi disk lokal.

Enkripsi data memerlukan kunci dan sertifikat. Konfigurasi keamanan memberi Anda fleksibilitas untuk memilih dari beberapa opsi, termasuk kunci yang dikelola oleh AWS Key Management Service, kunci yang dikelola oleh Amazon S3, dan kunci dan sertifikat dari penyedia kustom yang Anda suplai. Saat menggunakan AWS KMS sebagai penyedia kunci Anda, biaya berlaku untuk penyimpanan dan penggunaan kunci enkripsi. Untuk informasi lebih lanjut, lihat AWS KMS harga.

Sebelum Anda menentukan opsi enkripsi, tentukan kunci dan sistem pengelolaan sertifikat yang ingin Anda gunakan, sehingga Anda dapat terlebih dahulu membuat kunci dan sertifikat atau penyedia kustom yang Anda tentukan sebagai bagian dari pengaturan enkripsi.

Enkripsi saat istirahat untuk data EMRFS di Amazon S3

Enkripsi Amazon S3 bekerja dengan objek EMR File System (EMRFS) yang dibaca dari dan ditulis ke Amazon S3. Anda menentukan Amazon S3 server-side encryption (SSE) atau client-side encryption (CSE) sebagai Mode enkripsi default saat Anda mengaktifkan enkripsi saat istirahat. Secara opsional, Anda dapat menentukan metode enkripsi yang berbeda untuk setiap bucket menggunakan Per penimpaan enkripsi bucket. Keamanan Lapisan Pengangkutan (TLS) terlepas dari apakah enkripsi Amazon S3 diaktifkan, Keamanan Lapisan Pengangkutan (TLS) mengenkripsi objek EMRFS dalam transit antara simpul klaster EMR dan Amazon S3. Untuk informasi mendalam tentang enkripsi Amazon S3, lihat Melindungi data menggunakan enkripsi di Panduan Pengguna Layanan Penyimpanan Sederhana Amazon.

> **Note**
>
> Saat Anda menggunakan AWS KMS, biaya berlaku untuk penyimpanan dan penggunaan kunci enkripsi. Untuk informasi lebih lanjut, lihat AWS KMS Harga.

Enkripsi sisi server Amazon S3

Saat Anda mengatur enkripsi sisi server Amazon S3, Amazon S3 akan mengenkripsi data pada tingkat objek saat menulis data ke disk dan mendekripsi data saat diakses. Untuk informasi selengkapnya tentang SSE, lihat Melindungi data menggunakan enkripsi sisi server di Panduan Pengguna Layanan Penyimpanan Sederhana Amazon.

Anda dapat memilih di antara dua sistem manajemen kunci yang berbeda saat Anda menentukan SSE di Amazon EMR:

- SSE-S3 – Amazon S3 mengelola kunci untuk Anda.
- SSE-KMS - Anda menggunakan AWS KMS key untuk mengatur dengan kebijakan yang sesuai untuk Amazon EMR. Untuk informasi selengkapnya tentang persyaratan utama untuk Amazon EMR, lihat Menggunakan AWS KMS keys untuk enkripsi.

SSE dengan kunci yang disediakan pelanggan (SSE-C) tidak tersedia untuk digunakan dengan Amazon EMR.

Enkripsi di sisi klien Amazon S3

Dengan enkripsi sisi klien Amazon S3, enkripsi dan dekripsi Amazon S3 dilakukan di klien EMRFS pada klaster Anda. Objek dienkripsi sebelum diunggah ke Amazon S3 dan didekripsi setelah diunduh. Penyedia yang Anda tentukan menyediakan kunci enkripsi yang digunakan klien. Klien dapat menggunakan kunci yang disediakan oleh AWS KMS (CSE-KMS) atau kelas Java kustom yang menyediakan kunci root sisi klien (CSE-C). Spesifikasi enkripsi sedikit berbeda antara CSE-KMS dan CSE-C, tergantung pada penyedia yang ditentukan dan metadata objek yang didekripsi atau dienkripsi. Untuk informasi selengkapnya tentang perbedaan ini, lihat Melindungi data menggunakan enkripsi sisi klien di Panduan Pengguna Layanan Penyimpanan Sederhana Amazon.

> **Note**
>
> Amazon S3 CSE hanya memastikan bahwa data EMRFS yang dipertukarkan dengan Amazon S3 dienkripsi; tidak semua data pada volume instans klaster dienkripsi. Selain itu, karena Hue tidak menggunakan EMRFS, objek yang Hue S3 File Browser tulis ke Amazon S3 tidak dienkripsi.

Enkripsi disk lokal

Mekanisme berikut bekerja sama untuk mengenkripsi disk lokal ketika Anda mengaktifkan enkripsi disk lokal menggunakan konfigurasi keamanan Amazon EMR.

Enkripsi HDFS sumber terbuka

HDFS mempertukarkan data antara instans klaster selama pemrosesan terdistribusi. Hal ini juga membaca dari dan menulis data ke volume penyimpanan dan volume EBS terlampir ke instans. Opsi enkripsi Hadoop sumber terbuka berikut diaktifkan ketika Anda mengaktifkan enkripsi disk lokal:

- Secure Hadoop RPC diatur ke `Privacy`, yang menggunakan Simple Authentication Security Layer (SASL) sederhana.
- Enscriptsi data pada pemindahan data blok HDFS diatur ke `true` dan dikonfigurasi untuk menggunakan enkripsi AES 256.

> **Note**
>
> Anda dapat mengaktifkan enkripsi Apache Hadoop tambahan dengan mengaktifkan enkripsi dalam transit. Untuk informasi selengkapnya, lihat Enkripsi dalam transit. Pengaturan enkripsi ini tidak mengaktifkan enkripsi transparan HDFS, yang dapat Anda konfigurasikan secara manual. Untuk informasi selengkapnya, lihat Enkripsi transparan di HDFS di Amazon EMR di Panduan Amazon EMR rilis.

Enkripsi penyimpanan instans

Untuk tipe instans EC2 yang menggunakan SSD berbasis NVME sebagai volume penyimpanan instans, enkripsi NVMe digunakan terlepas dari pengaturan enkripsi Amazon EMR. Untuk informasi selengkapnya, lihat tipe volume Amazon EBS di Panduan Pengguna Amazon EC2 untuk Instans Linux. Untuk volume penyimpanan instans lain, Amazon EMR menggunakan LUKS untuk mengenkripsi volume penyimpanan instans ketika enkripsi disk lokal diaktifkan terlepas dari apakah volume EBS dienkripsi menggunakan enkripsi EBS atau LUKS.

Enkripsi volume EBS

Jika Anda membuat klaster di Wilayah tempat enkripsi Amazon EC2 volume EBS diaktifkan secara default untuk akun Anda, volume EBS dienkripsi meskipun enkripsi disk lokal tidak diaktifkan. Untuk

informasi lebih lanjut, lihat Enkripsi oleh default di Panduan Pengguna Amazon EC2 untuk Instans Linux. Dengan enkripsi disk lokal diaktifkan dalam konfigurasi keamanan, pengaturan EMR Amazon lebih diutamakan daripada pengaturan Amazon EC2 untuk instans EC2 cluster encryption-by-default .

Pilihan berikut tersedia untuk mengenkripsi volume EBS menggunakan konfigurasi keamanan:

- Enkripsi EBS – Dimulai dengan Amazon EMR versi 5.24.0, Anda dapat memilih untuk mengaktifkan enkripsi EBS. Opsi enkripsi EBS mengenkripsi volume perangkat asal EBS dan volume penyimpanan terlampir. Opsi enkripsi EBS tersedia hanya bila Anda menentukan AWS Key Management Service sebagai penyedia kunci Anda. Kami merekomendasikan penggunaan enkripsi EBS.
- Enkripsi LUKS Jika Anda memilih untuk menggunakan enkripsi LUKS untuk volume Amazon EBS, enkripsi LUKS hanya berlaku untuk volume penyimpanan terlampir, bukan ke volume perangkat asal. Untuk informasi selengkapnya tentang enkripsi LUKS, lihat spesifikasi pada disk LUKS.

  Untuk penyedia kunci Anda, Anda dapat menyiapkan kebijakan AWS KMS key dengan yang sesuai untuk Amazon EMR, atau kelas Java kustom yang menyediakan artefak enkripsi. Saat Anda menggunakan AWS KMS, biaya berlaku untuk penyimpanan dan penggunaan kunci enkripsi. Untuk informasi lebih lanjut, lihat AWS KMS harga.

> **Note**
>
> Untuk memeriksa apakah enkripsi EBS diaktifkan pada klaster Anda, dianjurkan bahwa Anda menggunakan `DescribeVolumes` panggilan API. Untuk informasi lebih lanjut, lihat DescribeVolumes. Menjalankan `lsblk` di klaster hanya akan memeriksa status enkripsi LUKS, bukan enkripsi EBS.

Enkripsi dalam transit

Beberapa mekanisme enkripsi diaktifkan dengan enkripsi dalam transit. Ini adalah fitur sumber daya terbuka, aplikasi-spesifik, dan dapat bervariasi dengan Amazon EMR rilis. Fitur enkripsi khusus aplikasi berikut dapat diaktifkan menggunakan konfigurasi aplikasi Apache. Untuk informasi selengkapnya, lihat Mengkonfigurasi aplikasi.

Hadoop

- Shuffle MapReduce terenkripsi Hadoop menggunakan TLS.

- Secure Hadoop RPC diatur ke "Privasi" dan menggunakan SASL (diaktifkan di Amazon EMR ketika enkripsi saat istirahat diaktifkan).
- Penyulitan data pada pemindahan data blok HDFS menggunakan AES 256 (diaktifkan di Amazon EMR saat enkripsi saat istirahat diaktifkan di konfigurasi keamanan).
- Untuk informasi selengkapnya, lihat Hadoop dalam mode aman di dokumentasi Apache Hadoop.

HBase

- Ketika Kerberos diaktifkan, properti `hbase.rpc.protection` diatur ke `privacy` untuk komunikasi terenkripsi.
- Untuk informasi selengkapnya, lihat Konfigurasi sisi klien untuk pengoperasian yang aman di dokumentasi Apache HBase.
- Untuk informasi lebih lanjut tentang Kerberos dengan Amazon EMR, lihat Gunakan Kerberos untuk otentikasi dengan Amazon EMR.

Hive

- Komunikasi klien JDBC/ODBC dengan HiveServer 2 (HS2) dienkripsi menggunakan konfigurasi SSL di Amazon EMR rilis 6.9.0 dan yang lebih baru.
- Untuk informasi selengkapnya, lihat bagian enkripsi SSL dari dokumentasi Apache Hive.

Spark

- Komunikasi RPC internal antara komponen Spark, seperti layanan transfer blok dan layanan shuffle eksternal, dienkripsi menggunakan cipher AES-256 di Amazon EMR versi 5.9.0 dan versi terbaru. Di rilis sebelumnya, komunikasi RPC internal dienkripsi menggunakan SASL dengan DIGEST-MD5 sebagai cipher.
- Komunikasi protokol HTTP dengan antarmuka pengguna seperti Spark History Server dan server file HTTPS-enabled dienkripsi menggunakan konfigurasi SSL Spark. Untuk informasi lebih lanjut, lihat Konfigurasi SSL di dokumentasi Spark.
- Untuk informasi selengkapnya, lihat bagian Pengaturan keamanan Spark pada dokumentasi Apache Spark.

Tez

- Tez shuffle handler menggunakan TLS (`tez.runtime.ssl.enable`).

Presto

- Komunikasi internal antara simpul Presto menggunakan SSL/TLS (hanya Amazon EMR versi 5.6.0 dan versi yang lebih baru).

Anda menentukan artefak enkripsi yang digunakan untuk enkripsi dalam transit di salah satu dari dua cara: baik dengan menyediakan file zip sertifikat yang Anda upload ke Amazon S3, atau dengan referensi kelas Java kustom yang menyediakan artefak enkripsi. Untuk informasi selengkapnya, lihat Memberikan sertifikat untuk mengenkripsi data dalam transit dengan enkripsi Amazon EMR.

Membuat kunci dan sertifikat untuk enkripsi data

Sebelum Anda menentukan opsi enkripsi menggunakan konfigurasi keamanan, putuskan penyedia yang ingin Anda gunakan untuk kunci dan artefak enkripsi. Misalnya, Anda dapat menggunakan AWS KMS atau penyedia kustom yang Anda buat. Selanjutnya, membuat kunci atau penyedia kunci seperti yang dijelaskan di bagian ini.

Menyediakan kunci untuk mengenkripsi data at rest dengan Amazon EMR

Anda dapat menggunakan AWS Key Management Service (AWS KMS) atau penyedia kunci kustom untuk enkripsi data di Amazon EMR. Saat Anda menggunakan AWS KMS, biaya berlaku untuk penyimpanan dan penggunaan kunci enkripsi. Untuk informasi lebih lanjut, lihat AWS KMS harga.

Topik ini memberikan detail kebijakan utama untuk kunci KMS yang akan digunakan dengan Amazon EMR, serta pedoman dan contoh kode untuk menulis kelas penyedia kunci khusus untuk enkripsi Amazon S3. Untuk informasi selengkapnya tentang pembuatan kunci, lihat Membuat Kunci di AWS Key Management Service Panduan Developer.

Menggunakan AWS KMS keys untuk enkripsi

Kunci enkripsi AWS KMS harus dibuat di Wilayah yang sama dengan instans klaster Amazon EMR Anda dan bucket Amazon S3 yang digunakan dengan EMRFS. Jika kunci yang Anda tentukan berada di akun yang berbeda dari yang Anda gunakan untuk mengkonfigurasi klaster, Anda harus menentukan kunci menggunakan ARN-nya.

Peran untuk profil instans Amazon EC2 harus memiliki izin untuk menggunakan kunci KMS yang Anda tentukan. Peran default untuk profil instans di Amazon EMR adalah `EMR_EC2_DefaultRole`. Jika Anda menggunakan peran yang berbeda untuk profil instans, atau Anda menggunakan IAM role untuk permintaan EMRFS ke Amazon S3, pastikan bahwa setiap peran ditambahkan sebagai pengguna kunci sebagaimana mestinya. Ini memberikan izin peran untuk menggunakan kunci KMS. Untuk informasi selengkapnya, lihat Menggunakan Kebijakan Utama di Panduan AWS Key Management Service Pengembang dan Mengonfigurasi peran IAM untuk permintaan EMRFS ke Amazon S3.

Anda dapat menggunakan AWS Management Console untuk menambahkan profil instans atau profil instans EC2 ke daftar pengguna utama untuk kunci KMS yang ditentukan, atau Anda dapat menggunakan AWS CLI atau AWS SDK untuk melampirkan kebijakan kunci yang sesuai.

Perhatikan bahwa Amazon EMR hanya mendukung kunci KMS simetris. Anda tidak dapat menggunakan kunci KMS asimetris untuk mengenkripsi data saat istirahat di cluster EMR Amazon. Untuk bantuan menentukan apakah kunci KMS simetris atau asimetris, lihat Mengidentifikasi kunci KMS simetris dan asimetris.

Prosedur di bawah ini menjelaskan cara menambahkan profil instans EMR Amazon default, `EMR_EC2_DefaultRole` sebagai pengguna utama yang menggunakan. AWS Management Console Ini mengasumsikan bahwa Anda telah membuat kunci KMS. Untuk membuat kunci KMS baru, lihat Membuat Kunci di Panduan AWS Key Management Service Pengembang.

Untuk menambahkan profil instans EC2 untuk Amazon EMR ke daftar pengguna kunci enkripsi

1. Masuk ke AWS Management Console lalu buka konsol AWS Key Management Service (AWS KMS) di https://console.aws.amazon.com/kms.
2. Untuk mengubah Wilayah AWS, gunakan pemilih Wilayah di sudut kanan atas halaman.
3. Pilih alias tombol KMS untuk memodifikasi.
4. Pada halaman detail kunci di bawah Pengguna Kunci, pilih Tambahkan.
5. Di kotak dialog Tambah pengguna kunci, pilih peran yang sesuai. Nama peran default adalah `EMR_EC2_DefaultRole`.
6. Pilih Tambahkan.

Mengaktifkan enkripsi EBS dengan memberikan izin tambahan untuk kunci KMS

Dimulai dengan Amazon EMR versi 5.24.0, Anda dapat mengenkripsi perangkat root EBS dan volume penyimpanan dengan menggunakan opsi konfigurasi keamanan. Untuk mengaktifkan opsi

tersebut, Anda harus menentukan AWS KMS sebagai penyedia kunci Anda. Selain itu, Anda harus memberikan peran layanan EMR `EMR_DefaultRole` dengan izin untuk menggunakan AWS KMS key yang Anda tentukan.

Anda dapat menggunakan AWS Management Console untuk menambahkan peran layanan EMR ke daftar pengguna kunci untuk kunci KMS yang ditentukan, atau Anda dapat menggunakan atau AWS SDK untuk melampirkan kebijakan kunci yang sesuai. AWS CLI

Prosedur di bawah ini menerangkan cara menambah peran layanan EMR default, `EMR_DefaultRole` sebagai pengguna kunci menggunakan AWS Management Console. Ini mengasumsikan bahwa Anda telah membuat kunci KMS. Untuk membuat kunci KMS baru, lihat Membuat kunci di Panduan AWS Key Management Service Pengembang.

Untuk menambahkan peran layanan EMR ke daftar pengguna kunci enkripsi

1. Masuk ke AWS Management Console dan buka AWS Key Management Service (AWS KMS) konsol di https://console.aws.amazon.com/kms.
2. Untuk mengubah Wilayah AWS, gunakan pemilih Wilayah di sudut kanan atas halaman.
3. Pilih Kunci terkelola pelanggan di bilah sisi kiri.
4. Pilih alias tombol KMS untuk memodifikasi.
5. Pada halaman detail kunci di bawah Pengguna Kunci, pilih Tambahkan.
6. Di kotak dialog Tambah pengguna kunci memilih peran yang sesuai. Nama peran layanan EMR default adalah `EMR_DefaultRole`.
7. Memilih Tambahkan.

Membuat penyedia kunci kustom

Bila menggunakan konfigurasi keamanan, Anda harus menentukan nama kelas penyedia yang berbeda untuk enkripsi disk lokal dan enkripsi Amazon S3.

Saat Anda membuat penyedia kunci khusus, aplikasi diharapkan mengimplementasikan EncryptionMaterialsProvider antarmuka, yang tersedia dalam AWS SDK for Java versi 1.11.0 dan yang lebih baru. Implementasinya dapat menggunakan strategi apapun untuk menyediakan materi enkripsi. Misalnya, Anda dapat memilih untuk menyediakan materi enkripsi statis atau mengintegrasikan dengan sistem manajemen kunci yang lebih kompleks.

Algoritma enkripsi yang digunakan untuk bahan enkripsi khusus harus NoPaddingAES/GCM/.

EncryptionMaterialsProvider Kelas mendapatkan materi enkripsi dengan konteks enkripsi. Amazon EMR mengisi informasi konteks enkripsi pada waktu aktif untuk membantu pemanggil dalam menentukan materi enkripsi yang benar untuk dikembalikan.

Example Contoh: Menggunakan penyedia kunci khusus untuk enkripsi Amazon S3 dengan EMRFS

Saat Amazon EMR mengambil materi enkripsi dari EncryptionMaterialsProvider kelas untuk melakukan enkripsi, EMRFS secara opsional mengisi argumen MaterialsDescription dengan dua bidang: URI Amazon S3 untuk objek dan cluster, yang dapat digunakan oleh kelas untuk mengembalikan materi enkripsi secara JobFlowId selektif. EncryptionMaterialsProvider

Misalnya, penyedia dapat mengembalikan kunci yang berbeda untuk prefiks URI Amazon S3 yang berbeda. Ini adalah deskripsi materi enkripsi yang dikembalikan yang pada akhirnya disimpan dengan objek Amazon S3 bukan nilai materialsDescription yang dihasilkan oleh EMRFS dan diteruskan ke penyedia. Saat mendekripsi objek Amazon S3, deskripsi bahan enkripsi diteruskan ke EncryptionMaterialsProvider kelas, sehingga dapat, sekali lagi, secara selektif mengembalikan kunci yang cocok untuk mendekripsi objek.

Implementasi EncryptionMaterialsProvider referensi disediakan di bawah ini. Penyedia kustom lain, EMRFSRSAEncryptionMaterialsProvider, tersedia dari. GitHub

```
import com.amazonaws.services.s3.model.EncryptionMaterials;
import com.amazonaws.services.s3.model.EncryptionMaterialsProvider;
import com.amazonaws.services.s3.model.KMSEncryptionMaterials;
import org.apache.hadoop.conf.Configurable;
import org.apache.hadoop.conf.Configuration;

import java.util.Map;

/**
 * Provides KMSEncryptionMaterials according to Configuration
 */
public class MyEncryptionMaterialsProviders implements EncryptionMaterialsProvider,
 Configurable{
  private Configuration conf;
  private String kmsKeyId;
  private EncryptionMaterials encryptionMaterials;

  private void init() {
    this.kmsKeyId = conf.get("my.kms.key.id");
    this.encryptionMaterials = new KMSEncryptionMaterials(kmsKeyId);
```

```
  }

  @Override
  public void setConf(Configuration conf) {
    this.conf = conf;
    init();
  }

  @Override
  public Configuration getConf() {
    return this.conf;
  }

  @Override
  public void refresh() {

  }

  @Override
  public EncryptionMaterials getEncryptionMaterials(Map<String, String>
 materialsDescription) {
    return this.encryptionMaterials;
  }

  @Override
  public EncryptionMaterials getEncryptionMaterials() {
    return this.encryptionMaterials;
  }
}
```

Memberikan sertifikat untuk mengenkripsi data dalam transit dengan enkripsi Amazon EMR

Dengan Amazon EMR rilis versi 4.8.0 atau yang lebih baru, Anda memiliki dua pilihan untuk menentukan artefak untuk mengenkripsi data dalam transit menggunakan konfigurasi keamanan:

- Anda dapat secara manual membuat sertifikat PEM, termasuk mereka di file .zip, dan kemudian referensi file .zip di Amazon S3.
- Anda dapat menerapkan penyedia sertifikat kustom sebagai kelas Java. Anda menentukan file JAR dari aplikasi di Amazon S3, dan lalu memberikan nama kelas lengkap dari penyedia seperti yang dinyatakan di aplikasi. Kelas harus mengimplementasikan ArtifactsProvider antarmuka TLS yang tersedia dimulai dengan AWS SDK for Java versi 1.11.0.

Amazon EMR secara otomatis mengunduh artefak ke setiap simpul di klaster dan versi terbaru menggunakannya untuk menerapkan fitur enkripsi dalam transit sumber daya terbuka. Untuk informasi lebih lanjut tentang menambahkan opsi, lihat Enkripsi dalam transit.

Menggunakan sertifikat PEM

Ketika Anda menetapkan file .zip untuk enkripsi dalam transit, konfigurasi keamanan mengharapkan file PEM di file .zip harus diberi nama persis seperti yang muncul di bawah ini:

Sertifikat enkripsi dalam transit

Nama file	Diperlukan/opsional	Detail
privateKey.pem	Diperlukan	Kunci privat
certificateChain.pem	Diperlukan	Rantai sertifikat
trustedCertificates.pem	Opsional	Diperlukan jika sertifikat yang disediakan tidak ditandatangani oleh salah satu otoritas sertifikasi (CA) root yang dipercaya default Java atau CA menengah yang dapat menautkan ke CA root yang dipercaya default Java. CA root yang dipercaya default Java dapat ditemukan di `jre/lib/security/cacerts` .

Anda mungkin ingin mengonfigurasi file PEM kunci privat menjadi sertifikat wildcard yang mengizinkan akses ke domain Amazon VPC di mana instans klaster Anda berada. Misalnya, jika klaster Anda berada di us-east-1 (N. Virginia), Anda bisa menentukan nama umum di konfigurasi sertifikat yang mengizinkan akses ke klaster dengan menentukan `CN=*.ec2.internal` di definisi subjek sertifikat. Jika klaster Anda berada di us-west-2 (Oregon), Anda dapat menentukan `CN=*.us-west-2.compute.internal`.

Jika file PEM disediakan di artefak enkripsi tidak memiliki karakter wildcard di CN untuk domain, Anda harus mengubah nilai `hadoop.ssl.hostname.verifier` ke `ALLOW_ALL`. Hal ini dilakukan

dengan klasifikasi `core-site` ketika mengirimkan konfigurasi ke klaster atau dengan menambahkan nilai ini di file `core-site.xml`. Perubahan ini diperlukan karena verifier nama host default tidak akan menerima nama host tanpa wildcard, yang akan mengakibatkan kesalahan. Untuk informasi selengkapnya tentang konfigurasi klaster EMR dalam VPC Amazon, lihat. Mengkonfigurasi jaringan

Contoh berikut menunjukkan cara menggunakan OpenSSL untuk membuat sertifikat X.509 yang bertandatangan sendiri dengan kunci privat RSA 1024-bit. Kunci ini memungkinkan akses ke instance cluster Amazon EMR penerbit di `us-west-2` Wilayah (Oregon) sebagaimana ditentukan oleh nama domain sebagai *`*.us-west-2.compute.internal`* nama umum.

Item subjek opsional lainnya, seperti negara (C), negara bagian (S), dan Lokal (L), ditentukan. Karena sertifikat yang bertandatangan sendiri dibuat, perintah kedua di contoh salinan `certificateChain.pem` file ke `trustedCertificates.pem` file. Perintah ketiga menggunakan `zip` untuk membuat file `my-certs.zip` yang berisi sertifikat.

> **Important**
>
> Contoh ini hanya proof-of-concept demonstrasi. Menggunakan sertifikat yang bertandatangan sendiri tidak direkomendasikan dan menimbulkan risiko keamanan potensial. Untuk sistem produksi, gunakan otoritas sertifikasi (CA) untuk menerbitkan sertifikat.

```
$ openssl req -x509 -newkey rsa:1024 -keyout privateKey.pem -out certificateChain.pem
 -days 365 -nodes -subj '/C=US/ST=Washington/L=Seattle/O=MyOrg/OU=MyDept/CN=*.us-
west-2.compute.internal'
$ cp certificateChain.pem trustedCertificates.pem
$ zip -r -X my-certs.zip certificateChain.pem privateKey.pem trustedCertificates.pem
```

AWS Identity and Access Management untuk Amazon EMR

(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 diotorisasi (memiliki izin) untuk menggunakan sumber daya Amazon EMR. IAM adalah sebuah layanan Layanan AWS yang dapat Anda gunakan tanpa dikenakan biaya tambahan.

Topik

- Audiens

- Mengautentikasi menggunakan identitas
- Mengelola kebijakan menggunakan akses
- Cara kerja Amazon EMR dengan IAM
- Peran runtime untuk langkah-langkah EMR Amazon
- Konfigurasi peran layanan IAM untuk izin Amazon EMR untuk layanan AWS dan sumber daya
- Kebijakan contoh berbasis identitas Amazon EMR.

Audiens

Cara Anda menggunakan AWS Identity and Access Management (IAM) berbeda, tergantung pada pekerjaan yang Anda lakukan di Amazon EMR.

Pengguna layanan – Jika Anda menggunakan layanan Amazon EMR untuk melakukan tugas Anda, administrator Anda akan memberikan kredensial dan izin yang dibutuhkan. Saat Anda menggunakan lebih banyak fitur Amazon EMR untuk melakukan tugas, Anda mungkin memerlukan izin tambahan. Memahami cara akses dikelola dapat membantu Anda meminta izin yang tepat dari administrator Anda. Jika Anda tidak dapat mengakses fitur di Amazon EMR, lihat. Memecahkan masalah identitas dan akses EMR Amazon

Administrator layanan – Jika Anda bertanggung jawab atas sumber daya Amazon EMR di korporasi Anda, Anda mungkin memiliki akses penuh ke Amazon EMR. Tugas Anda adalah menentukan fitur dan sumber daya Amazon EMR mana yang harus diakses pengguna layanan Anda. Kemudian, Anda harus mengirimkan permintaan kepada administrator IAM Anda untuk mengubah izin pengguna layanan Anda. Tinjau informasi di halaman ini untuk memahami konsep Basic IAM. Pelajari selengkapnya tentang cara korporasi Anda dapat menggunakan IAM dengan Amazon EMR, lihat Cara kerja Amazon EMR dengan IAM.

Administrator IAM – Jika Anda adalah administrator IAM, Anda mungkin ingin belajar dengan lebih terpedetail tentang cara Anda dapat menulis kebijakan untuk mengelola akses ke Amazon EMR. Untuk melihat kebijakan contoh berbasis identitas Amazon EMR yang dapat Anda gunakan di IAM, lihat Kebijakan contoh berbasis identitas Amazon EMR..

Mengautentikasi menggunakan identitas

Autentikasi merupakan cara Anda untuk masuk ke AWS dengan menggunakan kredensial identitas Anda. Anda harus terautentikasi (masuk keAWS) sebagai Pengguna root akun AWS, sebagai pengguna IAM, atau dengan mengambil peran IAM.

Anda dapat masuk ke AWS sebagai identitas terfederasi dengan menggunakan kredensial yang disediakan melalui sumber identitas. AWS IAM Identity Center Para pengguna (Pusat Identitas IAM), otentikasi sign-on tunggal perusahaan Anda, dan kredensial Google atau Facebook Anda merupakan contoh identitas terfederasi. Saat Anda masuk sebagai identitas terfederasi, administrator Anda sebelumnya menyiapkan federasi identitas dengan menggunakan peran IAM. Ketika Anda mengakses AWS dengan menggunakan federasi, Anda secara tidak langsung mengambil suatu peran.

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

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

Terlepas dari metode autentikasi yang Anda gunakan, Anda mungkin juga diminta untuk menyediakan informasi keamanan tambahan. Sebagai contoh, AWS menyarankan supaya Anda menggunakan autentikasi multi-faktor (MFA) untuk meningkatkan keamanan akun Anda. Untuk mempelajari selengkapnya, silakan lihat Autentikasi multi-faktor di Panduan Pengguna AWS IAM Identity Center dan Menggunakan autentikasi multi-faktor (MFA) di AWS di Panduan Pengguna IAM.

Pengguna root Akun AWS

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

Identitas terfederasi

Praktik terbaiknya berupa, mewajibkan pengguna manusia, termasuk pengguna yang memerlukan akses administrator, untuk menggunakan federasi dengan penyedia identitas untuk mengakses Layanan AWS dengan menggunakan kredensial temporer.

Identitas terfederasi adalah pengguna dari direktori pengguna perusahaan Anda, penyedia identitas web, dikenal sebagai AWS Directory Service, direktori Pusat Identitas, atau pengguna mana pun yang mengakses Layanan AWS dengan menggunakan kredensial yang disediakan melalui sumber identitas. Ketika identitas terfederasi mengakses Akun AWS, identitas tersebut mengambil peran, dan peran memberikan kredensial temporer.

Untuk pengelolaan akses terpusat, kami sarankan Anda menggunakanAWS IAM Identity Center. Anda dapat membuat pengguna dan grup di Pusat Identitas IAM, atau Anda dapat menghubungkan dan menyinkronkan ke sekumpulan pengguna dan grup di sumber identitas Anda sendiri untuk digunakan di semua Akun AWS Anda dan aplikasi Anda. Untuk informasi tentang Pusat Identitas IAM, silakan lihat Apakah Pusat Identitas IAM itu? di User Guide AWS IAM Identity Center.

Pengguna dan Grup IAM

Pengguna IAM adalah identitas dalam Akun AWS Anda yang memiliki izin khusus untuk satu orang atau aplikasi. Apabila memungkinkan, kami menyarankan untuk mengandalkan pada kredensial temporer alih-alih membuat pengguna IAM yang memiliki kredensyal jangka panjang seperti kata sandi dan kunci akses. Namun, jika Anda memiliki kasus penggunaan khusus yang memerlukan kredensial jangka panjang dengan pengguna IAM, kami menyarankan Anda memutar kunci akses. Untuk informasi selengkapnya, silakan lihat Memutar kunci akses secara teratur untuk kasus penggunaan yang memerlukan kredensial jangka panjang di Panduan Pengguna IAM.

Grup IAM adalah identitas yang menerangkan secara spesifik kumpulan pengguna IAM. Anda tidak dapat masuk sebagai kelompok. Anda dapat menggunakan grup untuk menerangkan secara spesifik izin untuk beberapa pengguna sekaligus. Grup membuat izin lebih mudah dikelola untuk sekelompok besar pengguna. Sebagai contoh, Anda dapat memiliki grup yang diberi nama AdminIAM dan memberikan izin kepada grup tersebut untuk mengelola sumber daya IAM.

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

Peran IAM

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

IAM role dengan kredensial temporer berguna dalam situasi berikut:

- Akses pengguna gabungan – Untuk menetapkan izin ke sebuah identitas terfederasi, Anda harus membuat sebuah peran dan menentukan izin untuk peran tersebut. Ketika identitas gabungan terfederasi mengautentikasi, identitas tersebut terhubung dengan peran dan diberikan izin yang ditentukan oleh peran. Untuk informasi tentang peran-peran untuk federasi, silakan lihat Membuat sebuah peran untuk Penyedia Identitas pihak ketiga di Panduan Pengguna IAM. Jika Anda menggunakan Pusat Identitas IAM, Anda mengonfigurasi serangkain izin. Untuk mengontrol apa yang dapat diakses identitas Anda setelah identitas tersebut diautentikasi, Pusat Identitas IAM mengkorelasikan izin yang diatur ke peran dalam IAM. Untuk informasi tentang rangkaian izin, silakan lihat Rangkaian izin di Panduan Pengguna AWS IAM Identity Center.
- Izin pengguna IAM sementara – Pengguna atau peran IAM dapat mengambil peran IAM untuk sementara mengambil izin berbeda untuk tugas tertentu.
- Akses lintas akun – Anda dapat menggunakan peran IAM untuk mengizinkan seseorang (pengguna utama tepercaya) di akun berbeda untuk mengakses sumber daya yang ada di akun Anda. Peran adalah cara utama untuk memberikan akses lintas akun. Namun, pada beberapa Layanan AWS, Anda dapat melampirkan kebijakan secara langsung ke sumber daya (alih-alih menggunakan suatu peran sebagai proksi). Untuk mempelajari perbedaan antara kebijakan peran dan kebijakan berbasis sumber daya untuk akses lintas akun, silakan lihat Bagaimana peran IAM role berbeda dari kebijakan berbasis sumber daya dalam Panduan Pengguna IAM.
- Akses lintas layanan – Sebagian Layanan AWS menggunakan fitur di Layanan AWS lainnya. Sebagai contoh, ketika Anda melakukan panggilan dalam suatu layanan, lazim pada layanan tersebut menjalankan aplikasi di Amazon EC2 atau menyimpan objek di Amazon S3. Suatu layanan mungkin melakukan hal tersebut menggunakan izin pengguna utama panggilan, menggunakan peran layanan, atau peran tertaut layanan.
  - Sesi akses maju (FAS) – Ketika Anda menggunakan pengguna IAM atau peran IAM untuk melakukan tindakan-tindakan di AWS, Anda akan dianggap sebagai seorang pengguna utama. Ketika Anda menggunakan beberapa layanan, Anda mungkin melakukan sebuah tindakan yang

kemudian dilanjutkan oleh tindakan lain pada layanan yang berbeda. FAS menggunakan izin dari pengguna utama untuk memanggil Layanan AWS, yang dikombinasikan dengan Layanan AWS yang diminta untuk membuat pengajuan ke layanan hilir. Permintaan FAS hanya diajukan ketika sebuah layanan menerima pengajuan yang memerlukan interaksi dengan Layanan AWS lain atau sumber daya lain untuk diselesaikan. Dalam hal ini, Anda harus memiliki izin untuk melakukan kedua tindakan tersebut. Untuk detail kebijakan ketika mengajukan permintaan FAS, silakan lihat Meneruskan sesi akses.

- Peran layanan – Sebuah peran layanan adalah sebuah peran IAM yang dijalankan oleh suatu layanan untuk melakukan tindakan atas nama Anda. Administrator IAM dapat membuat, memodifikasi, dan menghapus peran layanan dari dalam IAM. Untuk informasi selengkapnya, silakan lihat Membuat sebuah peran untuk mendelegasikan izin ke Layanan AWS dalam Panduan pengguna IAM.
- Peran tertaut layanan – Peran tertaut layanan adalah tipe 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.

- Aplikasi yang berjalan di Amazon EC2 – Anda dapat menggunakan peran IAM untuk mengelola kredensial temporer untuk aplikasi yang berjalan di instans EC2 dan mengajukan permintaan AWS CLI atau API AWS. Cara ini lebih baik daripada menyimpan kunci akses dalam instans EC2. Untuk menugaskan sebuah peran AWS ke instans EC2 dan membuatnya tersedia untuk semua aplikasinya, Anda dapat membuat sebuah profil instans yang dilampirkan ke instans. Profil instans berisi peran dan memungkinkan program yang berjalan di instans EC2 untuk mendapatkan kredensial sementara. Untuk informasi selengkapnya, silakan lihat Menggunakan peran IAM untuk memberikan izin ke aplikasi yang berjalan di instans Amazon EC2 di Panduan Pengguna IAM.

Untuk mempelajari apakah kita harus menggunakan peran IAM atau pengguna IAM, silakan lihat Kapan harus membuat peran IAM (alih-alih pengguna) di Panduan Pengguna IAM.

Mengelola kebijakan menggunakan akses

Anda mengendalikan akses di AWS dengan membuat kebijakan dan melampirkannya ke identitas atau sumber daya AWS. Kebijakan adalah objek di AWS yang, ketika terkait dengan identitas atau sumber daya, akan menentukan izinnya. AWS mengevaluasi kebijakan-kebijakan tersebut ketika seorang pengguna utama (pengguna, root user, atau sesi peran) mengajukan permintaan. Izin dalam kebijakan menentukan apakah permintaan diberikan atau ditolak. Sebagian besar kebijakan disimpan

di AWS sebagai dokumen JSON. Untuk informasi selengkapnya tentang struktur dan isi dokumen kebijakan JSON, silakan lihat Gambaran Umum kebijakan JSON di Panduan Pengguna IAM.

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.

Secara bawaan, para pengguna dan peran tidak memiliki izin. 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.

Kebijakan IAM mendefinisikan izin untuk suatu tindakan terlepas dari metode yang Anda gunakan untuk pengoperasiannya. Sebagai contoh, anggap saja Anda memiliki kebijakan yang mengizinkan tindakan `iam:GetRole` . Pengguna dengan kebijakan tersebut dapat memperoleh informasi peran dari AWS Management Console, AWS CLI, atau APIAWS.

Kebijakan berbasis identitas

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.

Kebijakan berbasis identitas dapat dikategorikan lebih lanjut sebagai kebijakan inline atau kebijakan terkelola. Kebijakan inline ditanam secara langsung ke pengguna tunggal, grup, atau peran. Kebijakan terkelola adalah kebijakan yang berdiri sendiri yang dapat Anda lampirkan ke beberapa pengguna, grup, dan peran di Akun AWS Anda. Kebijakan terkelola mencakup kebijakan terkelola AWS dan kebijakan terkelola pelanggan. Untuk mempelajari cara memilih antara kebijakan terkelola atau kebijakan inline, silakan lihat Memilih antara kebijakan terkelola dan kebijakan inline diPanduan Pengguna IAM.

Kebijakan berbasis sumber daya

Kebijakan berbasis sumber daya adalah dokumen kebijakan JSON yang Anda lampirkan ke sumber daya. Contoh kebijakan-kebijakan berbasis sumber daya adalah kebijakan terpercaya peran IAM dan kebijakan bucket Amazon S3. Dalam layanan yang mendukung kebijakan berbasis sumber daya, administrator layanan dapat menggunakannya untuk mengontrol akses ke sumber daya tertentu.

Untuk sumber daya yang dilampiri kebijakan, kebijakan tersebut menentkan tindakan apa yang dapat dilakukan oleh pengguna utama yang ditentukan di sumber daya tersebut dan dalam kondisi apa. Anda harus menentukan pengguna utama dalam kebijakan berbasis sumber daya. Pengguna utama dapat mencakup akun, pengguna, peran, pengguna gabungan, atau Layanan AWS.

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

Daftar kontrol akses (ACL)

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

Amazon S3, AWS WAF, dan Amazon VPC adalah contoh-contoh layanan yang mendukung ACL. Untuk mempelajari ACL selengkapnya, silakan lihat Gambaran umum daftar kontrol akses (ACL) di Panduan Pengembang Layanan Penyimpanan Ringkas Amazon.

Tipe-tipe kebijakan lain

AWS mendukung tipe kebijakan tambahan, yang kurang umum. Tipe-tipe kebijakan ini dapat mengatur izin maksimum yang diberikan kepada Anda oleh tipe kebijakan yang lebih umum.

- Batasan izin – Batasan izin adalah fitur lanjutan tempat Anda mengatur izin maksimum yang dapat diberikan oleh kebijakan berbasis identitas kepada entitas IAM (pengguna IAM atau peran IAM). Anda dapat menetapkan batasan izin untuk suatu entitas. Izin yang dihasilkan adalah perpotongan antara kebijakan berbasis identitas milik entitas dan batas izinnya. Kebijakan berbasis sumber daya yang menentukan pengguna atau peran dalam bidang `Principal` tidak dibatasi oleh batasan izin. Penolakan eksplisit dalam salah satu kebijakan ini menindihi izin. Untuk informasi selengkapnya tentang batasan izin, silakan lihat Batasan izin untuk entitas IAM di Panduan Pengguna IAM.
- Kebijakan kontrol layanan (SCP) – SCP adalah kebijakan JSON yang menentukan izin maksimum untuk sebuah organisasi atau unit organisasi (OU) di AWS Organizations. AWS Organizations adalah layanan untuk mengelompokkan dan secara terpusat mengelola beberapa Akun AWS yang dimiliki bisnis Anda. Jika Anda mengaktifkan semua fitur di suatu organisasi, maka Anda dapat menerapkan kebijakan kontrol layanan (SCP) ke salah satu atau ke semua akun Anda. SCP membatasi izin untuk entitas dalam akun anggota, termasuk setiap Pengguna root akun AWS. Untuk informasi selengkapnya tentang Organisasi dan SCP, silakan lihat Cara kerja SCP di Panduan Pengguna AWS Organizations.

- Kebijakan sesi – Kebijakan sesi adalah kebijakan lanjutan yang Anda berikan sebagai parameter ketika Anda membuat sesi sementara secara terprogram untuk peran atau pengguna gabungan. Izin sesi yang dihasilkan adalah perpotongan kebijakan berbasis identitas pengguna atau peran dan kebijakan sesi. Izin juga dapat berasal dari kebijakan berbasis sumber daya. Penolakan eksplisit dalam salah satu kebijakan ini menindihi izin. Untuk informasi selengkapnya, silakan lihat Kebijakan sesi di Panduan Pengguna IAM.

Berbagai tipe kebijakan

Ketika beberapa tipe kebijakan berlaku pada suatu permintaan, izin yang dihasilkan lebih rumit untuk dipahami. Untuk mempelajari bagaimana AWS menentukan apakah mengizinkan permintaan jika beberapa tipe kebijakan dilibatkan, lihat Logika evaluasi kebijakan di Panduan Pengguna IAM.

Cara kerja Amazon EMR dengan IAM

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

Fitur IAM yang dapat Anda gunakan dengan Amazon EMR

Fitur IAM	Dukungan Amazon EMR
Kebijakan berbasis identitas	Ya
Kebijakan berbasis sumber daya	Ya
Tindakan kebijakan	Ya
Sumber daya kebijakan	Ya
Kunci kondisi kebijakan	Ya
ACL	Tidak
ABAC (tag dalam kebijakan)	Ya
Kredensial temporer	Ya
Izin-izin pengguna utama	Ya
Peran layanan	Tidak

Fitur IAM	Dukungan Amazon EMR
Peran tertaut layanan	Ya

Untuk mendapatkan tampilan tingkat tinggi tentang cara Amazon EMR dan layanan AWS lainnya bekerja dengan sebagian besar fitur IAM, AWSlihat layanan yang bekerja dengan IAM di Panduan Pengguna IAM.

Kebijakan berbasis identitas untuk Amazon EMR

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 Amazon EMR

Untuk melihat contoh identitas berbasis kebijakan Amazon EMR, lihat Kebijakan contoh berbasis identitas Amazon EMR..

Kebijakan berbasis sumber daya dalam Amazon EMR

Mendukung kebijakan berbasis sumber daya	Ya

Kebijakan berbasis sumber daya adalah dokumen kebijakan JSON yang Anda lampirkan ke sumber daya. Contoh kebijakan-kebijakan berbasis sumber daya adalah kebijakan terpercaya peran IAM dan

kebijakan bucket Amazon S3. Dalam layanan yang mendukung kebijakan berbasis sumber daya, administrator layanan dapat menggunakannya untuk mengontrol akses ke sumber daya tertentu. Untuk sumber daya yang dilampiri kebijakan, kebijakan tersebut menentkan tindakan apa yang dapat dilakukan oleh pengguna utama yang ditentukan di sumber daya tersebut dan dalam kondisi apa. Anda harus menentukan pengguna utama dalam kebijakan berbasis sumber daya. Pengguna utama dapat mencakup akun, pengguna, peran, pengguna gabungan, atau Layanan AWS.

Untuk mengaktifkan akses lintas akun, Anda dapat menentukan secara spesifik seluruh akun atau entitas IAM di akun lain sebagai pengguna utama dalam kebijakan berbasis sumber daya. Menambahkan pengguna utama akun silang ke kebijakan berbasis sumber daya hanya setengah dari membangun hubungan kepercayaan. Ketika pengguna utama dan sumber daya berada dalam Akun AWS yang berbeda, Administrator IAM di akun tepercaya juga harus memberikan izin kepada entitas pengguna utama (pengguna atau peran) untuk mengakses sumber daya. Mereka memberikan izin melampirkan kebijakan berbasis identitas kepada entitas. Namun, jika kebijakan berbasis sumber daya memberikan akses kepada pengguna utama dalam akun yang sama, tidak diperlukan kebijakan berbasis identitas tambahan. Untuk informasi selengkapnya, silakan lihat Bagaimana peran IAM berbeda dari kebijakan berbasis sumber daya di Panduan Pengguna IAM.

Tindakan kebijakan untuk Amazon EMR

Mendukung tindakan 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 `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.

Untuk melihat daftar tindakan EMR Amazon, lihat Tindakan, sumber daya, dan kunci kondisi untuk EMR Amazon di Referensi Otorisasi Layanan.

Tindakan kebijakan di Amazon EMR menggunakan awalan berikut sebelum tindakan:

```
EMR
```

Untuk menetapkan secara spesifik beberapa tindakan dalam satu pernyataan, pisahkan tindakan-tindakan tersebut dengan koma.

```
"Action": [
      "EMR:action1",
      "EMR:action2"
         ]
```

Untuk melihat contoh identitas berbasis kebijakan Amazon EMR, lihat Kebijakan contoh berbasis identitas Amazon EMR..

Sumber daya kebijakan untuk Amazon EMR

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": "*"
```

Untuk melihat daftar jenis sumber daya EMR Amazon dan ARNnya, lihat Sumber Daya yang Ditentukan oleh Amazon EMR di Referensi Otorisasi Layanan. Untuk mempelajari tindakan mana yang dapat Anda tentukan ARN dari setiap sumber daya, lihat Tindakan, sumber daya, dan kunci kondisi untuk Amazon EMR.

Untuk melihat contoh identitas berbasis kebijakan Amazon EMR, lihat Kebijakan contoh berbasis identitas Amazon EMR..

Kunci kondisi kebijakan untuk Amazon EMR

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.

Untuk melihat daftar kunci kondisi EMR Amazon dan untuk mempelajari tindakan dan sumber daya yang dapat Anda gunakan kunci kondisi, lihat Tindakan, sumber daya, dan kunci kondisi untuk Amazon EMR di Referensi Otorisasi Layanan.

Untuk melihat contoh identitas berbasis kebijakan Amazon EMR, lihat Kebijakan contoh berbasis identitas Amazon EMR..

Daftar kontrol akses (ACL) di Amazon EMR

Mendukung ACL	Tidak

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

Kontrol akses berbasis atribut (ABAC) dengan Amazon EMR

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.

Menggunakan kredensyal Sementara dengan Amazon EMR

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.

Izin utama lintas layanan untuk Amazon EMR

Mendukung sesi akses maju (FAS)	Ya

Saat Anda menggunakan pengguna IAM atau peran IAM untuk mengerjakan tindakan di AWS, Anda akan dianggap sebagai pengguna utama. Ketika Anda menggunakan beberapa layanan, Anda mungkin melakukan sebuah tindakan yang kemudian dilanjutkan oleh tindakan lain pada layanan yang berbeda. FAS menggunakan izin dari pengguna utama untuk memanggil Layanan AWS, yang dikombinasikan dengan Layanan AWS yang diminta untuk membuat pengajuan ke

layanan hilir. Permintaan FAS hanya diajukan ketika sebuah layanan menerima pengajuan yang memerlukan interaksi dengan Layanan AWS lain atau sumber daya lain untuk diselesaikan. Dalam hal ini, Anda harus memiliki izin untuk melakukan kedua tindakan tersebut. Untuk detail kebijakan ketika mengajukan permintaan FAS, silakan lihat Meneruskan sesi akses.

Peran layanan untuk Amazon EMR

Mendukung peran layanan	Tidak

Peran terkait layanan untuk Amazon EMR

Mendukung peran yang terhubung dengan layanan	Ya

Untuk informasi selengkapnya tentang cara membuat atau mengelola peran terkait layanan, lihat AWS layanan yang bekerja dengan IAM. Cari layanan dalam tabel yang memiliki `Yes` di kolom Service-linked role (Peran yang terhubung dengan layanan). Pilih tautan Ya untuk melihat dokumentasi peran tertaut layanan untuk layanan tersebut.

Gunakan klaster dan tanda Notebook dengan kebijakan IAM untuk kendali akses

Izin untuk tindakan Amazon EMR yang terkait dengan EMR Notebooks dan klaster EMR dapat disetel dengan baik menggunakan kendali akses berbasis tanda dengan kebijakan IAM berbasis identitas. Anda dapat menggunakan kunci syarat di `Condition` elemen (juga disebut blok `Condition`) untuk mengizinkan tindakan tertentu hanya ketika notebook, klaster, atau keduanya memiliki kunci tanda tertentu atau kombinasi kunci-nilai. Anda juga dapat membatasi `CreateEditor` tindakan (yang menciptakan EMR Notebooks) dan `RunJobFlow`tindakan (yang membuat sebuah klaster) sehingga permintaan untuk tanda harus disampaikan ketika sumber daya dibuat.

Di Amazon EMR, kunci syarat yang dapat digunakan di `Condition` elemen hanya berlaku untuk mereka tindakan API Amazon EMR di mana `ClusterID` atau `NotebookID` adalah parameter permintaan yang diperlukan. Misalnya, ModifyInstanceGroupstindakan tidak mendukung kunci konteks karena `ClusterID` merupakan parameter opsional.

Saat Anda membuat buku catatan EMR, tag default diterapkan dengan string kunci yang `creatorUserId` disetel ke nilai ID pengguna IAM yang membuat buku catatan. Ini berguna untuk membatasi tindakan yang diizinkan untuk notebook hanya untuk pencipta.

Kunci syarat berikut tersedia di Amazon EMR:

- Penggunaan `elasticmapreduce:ResourceTag/`*`TagKeyString`* kunci konteks syarat untuk mengizinkan atau menolak tindakan pengguna pada grup atau notebook dengan tanda yang memiliki *`TagKeyString`* yang Anda tentukan. Jika tindakan melewati kedua `ClusterID` dan `NotebookID`, syarat ini berlaku untuk klaster dan notebook. Ini berarti bahwa kedua sumber daya harus memiliki tanda kunci string atau kombinasi kunci-nilai yang Anda tentukan. Anda dapat menggunakan `Resource` elemen untuk membatasi pernyataan sehingga hanya berlaku untuk klaster atau notebook yang diperlukan. Untuk informasi selengkapnya, lihat Kebijakan contoh berbasis identitas Amazon EMR..
- Penggunaan `elasticmapreduce:RequestTag/`*`TagKeyString`* kunci konteks syarat untuk memerlukan tanda tertentu dengan panggilan tindakan/API. Misalnya, Anda dapat menggunakan kunci konteks syarat ini bersama dengan tindakan `CreateEditor` untuk mewajibkan bahwa sebuah kunci dengan *`TagKeyString`* yang diterapkan ke notebook saat dibuat.

Contoh

Untuk melihat daftar tindakan Amazon EMR, lihat Tindakan Ditetapkan oleh Amazon EMR di Panduan Pengguna IAM.

Peran runtime untuk langkah-langkah EMR Amazon

Peran runtime adalah peran AWS Identity and Access Management (IAM) yang dapat Anda tentukan saat mengirimkan pekerjaan atau kueri ke klaster EMR Amazon. Pekerjaan atau kueri yang Anda kirimkan ke klaster EMR Amazon menggunakan peran runtime untuk mengakses AWS sumber daya, seperti objek di Amazon S3. Anda dapat menentukan peran runtime dengan Amazon EMR untuk pekerjaan Spark dan Hive.

Anda juga dapat menentukan peran runtime saat tersambung ke kluster EMR Amazon Amazon SageMaker di dan saat Anda melampirkan Amazon EMR Studio Workspace ke kluster EMR. Untuk informasi selengkapnya, lihat Connect ke klaster EMR Amazon dari Studio dan. Jalankan EMR Studio Workspace dengan peran runtime

Sebelumnya, klaster EMR Amazon menjalankan pekerjaan atau kueri EMR Amazon dengan izin berdasarkan kebijakan IAM yang dilampirkan pada profil instance yang Anda gunakan untuk meluncurkan klaster. Ini berarti bahwa kebijakan harus berisi penyatuan semua izin untuk semua pekerjaan dan kueri yang berjalan di kluster EMR Amazon. Dengan peran runtime, Anda sekarang

dapat mengelola kontrol akses untuk setiap pekerjaan atau kueri satu per satu, alih-alih membagikan profil instans EMR Amazon pada klaster.

Di klaster EMR Amazon dengan peran runtime, Anda juga dapat menerapkan kontrol akses AWS Lake Formation berbasis ke pekerjaan dan kueri Spark, Hive, dan Presto terhadap data lake Anda. Untuk mempelajari lebih lanjut tentang cara mengintegrasikan denganAWS Lake Formation, lihatIntegrasi Amazon EMR dengan AWS Lake Formation.

> **Note**
>
> Bila Anda menentukan peran runtime untuk langkah EMR Amazon, lowongan atau kueri yang Anda kirimkan hanya dapat AWS mengakses sumber daya yang diizinkan oleh kebijakan yang dilampirkan pada peran runtime. Pekerjaan dan kueri ini tidak dapat mengakses Layanan Metadata Instans pada instans EC2 klaster atau menggunakan profil instans EC2 klaster untuk mengakses sumber daya apa pun. AWS

Prasyarat untuk meluncurkan cluster EMR Amazon dengan peran runtime

Topik

- Langkah 1: Siapkan konfigurasi keamanan di Amazon EMR
- Langkah 2: Siapkan profil instans EC2 untuk klaster EMR Amazon
- Langkah 3: Siapkan kebijakan kepercayaan

Langkah 1: Siapkan konfigurasi keamanan di Amazon EMR

Gunakan struktur JSON berikut untuk membuat konfigurasi keamanan pada AWS Command Line Interface (AWS CLI), dan atur `EnableApplicationScopedIAMRole` ke`true`. Untuk informasi selengkapnya tentang konfigurasi keamanan, lihat Menggunakan konfigurasi keamanan untuk mengatur keamanan klaster.

```
{
    "AuthorizationConfiguration":{
        "IAMConfiguration":{
            "EnableApplicationScopedIAMRole":true
        }
    }
}
```

Kami menyarankan agar Anda selalu mengaktifkan opsi enkripsi dalam transit dalam konfigurasi keamanan, sehingga data yang ditransfer melalui internet dienkripsi, bukan dalam teks biasa. Anda dapat melewati opsi ini jika Anda tidak ingin terhubung ke kluster EMR Amazon dengan peran runtime dari SageMaker Runtime Studio atau EMR Studio. Untuk mengonfigurasi enkripsi data, lihat Mengkonfigurasi enkripsi data.

Atau, Anda dapat membuat konfigurasi keamanan dengan pengaturan khusus dengan AWS Management Console.

Langkah 2: Siapkan profil instans EC2 untuk klaster EMR Amazon

Cluster EMR Amazon menggunakan peran profil instans Amazon EC2 untuk mengambil peran runtime. Untuk menggunakan peran runtime dengan langkah-langkah EMR Amazon, tambahkan kebijakan berikut ke peran IAM yang akan digunakan sebagai peran profil instance. Untuk menambahkan kebijakan ke peran IAM atau mengedit kebijakan sebaris atau terkelola yang ada, lihat Menambahkan dan menghapus izin identitas IAM.

```
{
    "Version":"2012-10-17",
    "Statement":[
        {
            "Sid":"AllowRuntimeRoleUsage",
            "Effect":"Allow",
            "Action":[
                "sts:AssumeRole",
                "sts:TagSession"
            ],
            "Resource":[
                <runtime-role-ARN>
            ]
        }
    ]
}
```

Langkah 3: Siapkan kebijakan kepercayaan

Untuk setiap peran IAM yang Anda rencanakan untuk digunakan sebagai peran runtime, tetapkan kebijakan kepercayaan berikut, ganti `EMR_EC2_DefaultRole` dengan peran profil instans Anda. Untuk mengubah kebijakan kepercayaan peran IAM, lihat Memodifikasi kebijakan kepercayaan peran.

```
{
    "Sid":"AllowAssumeRole",
    "Effect":"Allow",
    "Principal":{
        "AWS":"arn:aws:iam::<AWS_ACCOUNT_ID>:role/EMR_EC2_DefaultRole"
    },
    "Action":"sts:AssumeRole"
}
```

Luncurkan kluster EMR Amazon dengan kontrol akses berbasis peran

Setelah mengatur konfigurasi, Anda dapat meluncurkan klaster EMR Amazon dengan konfigurasi keamanan dari. Langkah 1: Siapkan konfigurasi keamanan di Amazon EMR Untuk menggunakan peran runtime dengan langkah-langkah EMR Amazon, gunakan `emr-6.7.0` label rilis atau versi lebih baru, dan pilih Hive, Spark, atau keduanya sebagai aplikasi cluster Anda. Untuk terhubung dari SageMaker Studio, gunakan rilis `emr-6.9.0` atau yang lebih baru, dan pilih Livy, Spark, Hive, atau Presto sebagai aplikasi cluster Anda. Untuk petunjuk tentang cara meluncurkan klaster Anda, lihatMenentukan konfigurasi keamanan untuk sebuah klaster.

Kirim pekerjaan Spark menggunakan langkah-langkah Amazon EMR

Berikut ini adalah contoh bagaimana menjalankan HdfsTest contoh yang disertakan dengan Apache Spark. Panggilan API ini hanya berhasil jika peran runtime Amazon EMR yang disediakan dapat mengakses. `S3_LOCATION`

```
RUNTIME_ROLE_ARN=<runtime-role-arn>
S3_LOCATION=<s3-path>
REGION=<aws-region>
CLUSTER_ID=<cluster-id>

aws emr add-steps --cluster-id $CLUSTER_ID \
--steps '[{ "Name": "Spark Example", "ActionOnFailure": "CONTINUE","HadoopJarStep":
 { "Jar":"command-runner.jar","Args" : ["spark-example","HdfsTest",
 "$S3_LOCATION"] } }]' \
--execution-role-arn $RUNTIME_ROLE_ARN \
--region $REGION
```

> **Note**
>
> Kami menyarankan Anda mematikan akses SSH ke kluster EMR Amazon dan hanya mengizinkan API `AddJobFlowSteps` EMR Amazon untuk mengakses ke cluster.

Kirim pekerjaan Hive menggunakan langkah-langkah EMR Amazon

Contoh berikut menggunakan Apache Hive dengan langkah-langkah Amazon EMR untuk mengirimkan pekerjaan untuk menjalankan file. `QUERY_FILE.hql` Kueri ini hanya berhasil jika peran runtime yang disediakan dapat mengakses jalur Amazon S3 dari file kueri.

```
RUNTIME_ROLE_ARN=<runtime-role-arn>
REGION=<aws-region>
CLUSTER_ID=<cluster-id>

aws emr add-steps --cluster-id $CLUSTER_ID \
--steps '[{ "Name": "Run hive query using command-runner.jar - simple
 select","ActionOnFailure":"CONTINUE","HadoopJarStep": { "Jar": "command-
runner.jar","Args" :["hive -
f","s3://DOC_EXAMPLE_BUCKET/QUERY_FILE.hql"] } }]' \
--execution-role-arn $RUNTIME_ROLE_ARN \
--region $REGION
```

Connect ke klaster EMR Amazon dengan peran runtime dari notebook Studio SageMaker

Anda dapat menerapkan peran runtime Amazon EMR ke kueri yang dijalankan di klaster EMR Amazon dari Studio. SageMaker Untuk melakukannya, lanjutkan langkah-langkah berikut.

1. Ikuti petunjuk di Luncurkan Amazon SageMaker Studio untuk membuat SageMaker Studio.
2. Di UI SageMaker Studio, mulai buku catatan dengan kernel yang didukung. Misalnya, mulai SparkMagic gambar dengan PySpark kernel.
3. Pilih klaster EMR Amazon di SageMaker Studio, lalu pilih Connect.
4. Pilih peran runtime, lalu pilih Connect.

Ini akan membuat sel SageMaker notebook dengan perintah ajaib untuk terhubung ke cluster EMR Amazon Anda dengan peran runtime Amazon EMR yang dipilih. Di sel notebook, Anda dapat memasukkan dan menjalankan kueri dengan peran runtime dan kontrol akses berbasis Lake

Formation. Untuk contoh lebih detail, lihat Menerapkan kontrol akses data berbutir halus dengan dan AWS Lake Formation Amazon EMR dari Amazon Studio. SageMaker

Kontrol akses ke peran runtime Amazon EMR

Anda dapat mengontrol akses ke peran runtime dengan tombol `elasticmapreduce:ExecutionRoleArn` kondisi. Kebijakan berikut memungkinkan prinsipal IAM untuk menggunakan peran IAM bernama`Caller`, atau peran IAM apa pun yang dimulai dengan string`CallerTeamRole`, sebagai peran runtime.

> **Important**
>
> Anda harus membuat kondisi berdasarkan kunci `elasticmapreduce:ExecutionRoleArn` konteks saat Anda memberikan akses pemanggil untuk memanggil `AddJobFlowSteps` atau `GetClusterSessionCredentials` API, seperti yang ditunjukkan contoh berikut.

```
{
    "Sid":"AddStepsWithSpecificExecRoleArn",
    "Effect":"Allow",
    "Action":[
        "elasticmapreduce:AddJobFlowSteps"
    ],
    "Resource":"*",
    "Condition":{
        "StringEquals":{
            "elasticmapreduce:ExecutionRoleArn":[
                "arn:aws:iam::<AWS_ACCOUNT_ID>:role/Caller"
            ]
        },
        "StringLike":{
            "elasticmapreduce:ExecutionRoleArn":[
                "arn:aws:iam::<AWS_ACCOUNT_ID>:role/CallerTeamRole*"
            ]
        }
    }
}
```

Membangun kepercayaan antara peran runtime dan klaster EMR Amazon

Amazon EMR menghasilkan pengenal unik `ExternalId` untuk setiap konfigurasi keamanan dengan otorisasi peran runtime yang diaktifkan. Otorisasi ini memungkinkan setiap pengguna untuk memiliki satu set peran runtime untuk digunakan pada cluster milik mereka. Misalnya, di perusahaan, setiap departemen dapat menggunakan ID eksternal mereka untuk memperbarui kebijakan kepercayaan pada rangkaian peran runtime mereka sendiri.

Anda dapat menemukan ID eksternal dengan Amazon EMR `DescribeSecurityConfiguration` API, seperti yang ditunjukkan pada contoh berikut.

```
aws emr describe-security-configuration --name 'iamconfig-with-lf'{"Name": "iamconfig-
with-lf",
    "SecurityConfiguration":
        "{\"AuthorizationConfiguration\":{\"IAMConfiguration\":
{\"EnableApplicationScopedIAMRole\
        ":true,\"ApplicationScopedIAMRoleConfiguration\":{\"PropagateSourceIdentity
\":true,\"Exter
        nalId\":\"FXH5TSACFDWUCDSR3YQE2O7ETPUSM4OBCGLYWODSCUZDNZ4Y\"}},\"Lake
        FormationConfiguration\":{\"AuthorizedSessionTagValue\":\"Amazon EMR\"}}}",
    "CreationDateTime": "2022-06-03T12:52:35.308000-07:00"
}
```

Untuk informasi tentang cara menggunakan ID eksternal, lihat Cara menggunakan ID eksternal saat memberikan akses ke AWS sumber daya Anda kepada pihak ketiga.

Audit

Untuk memantau dan mengontrol tindakan yang dilakukan pengguna akhir dengan peran IAM, Anda dapat mengaktifkan fitur identitas sumber. Untuk mempelajari lebih lanjut tentang identitas sumber, lihat Memantau dan mengontrol tindakan yang diambil dengan peran yang diasumsikan.

Untuk melacak identitas sumber, atur `ApplicationScopedIAMRoleConfiguration/PropagateSourceIdentity` ke `true` dalam konfigurasi keamanan Anda, sebagai berikut.

```
{
    "AuthorizationConfiguration":{
        "IAMConfiguration":{
            "EnableApplicationScopedIAMRole":true,
            "ApplicationScopedIAMRoleConfiguration":{
                "PropagateSourceIdentity":true
```

```
            }
        }
    }
}
```

Saat disetel `PropagateSourceIdentity` ke`true`, Amazon EMR menerapkan identitas sumber dari kredensyal panggilan ke sesi pekerjaan atau kueri yang Anda buat dengan peran runtime. Jika tidak ada identitas sumber yang ada dalam kredensyal panggilan, Amazon EMR tidak menyetel identitas sumber.

Untuk menggunakan properti ini, berikan `sts:SetSourceIdentity` izin ke profil instans Anda, sebagai berikut.

```
{ // PropagateSourceIdentity statement
    "Sid":"PropagateSourceIdentity",
    "Effect":"Allow",
    "Action":"sts:SetSourceIdentity",
    "Resource":[
        <runtime-role-ARN>
    ],
    "Condition":{
        "StringEquals":{
            "sts:SourceIdentity":<source-identity>
        }
    }
}
```

Anda juga harus menambahkan `AllowSetSourceIdentity` pernyataan ke kebijakan kepercayaan peran runtime Anda.

```
{ // AllowSetSourceIdentity statement
    "Sid":"AllowSetSourceIdentity",
    "Effect":"Allow",
    "Principal":{
        "AWS":"arn:aws:iam::<AWS_ACCOUNT_ID>:role/EMR_EC2_DefaultRole"
    },
    "Action":[
        "sts:SetSourceIdentity",
        "sts:AssumeRole"
    ],
    "Condition":{
        "StringEquals":{
```

```
            "sts:SourceIdentity":<source-identity>
        }
    }
}
```

Pertimbangan tambahan

> **Note**
>
> Dengan rilis Amazon EMR`emr-6.9.0`, Anda mungkin mengalami kegagalan intermiten saat terhubung ke klaster EMR Amazon dari Studio. SageMaker Untuk mengatasi masalah ini, Anda dapat menginstal tambalan dengan tindakan bootstrap saat meluncurkan cluster. Untuk detail tambalan, lihat Amazon EMR rilis 6.9.0 masalah yang diketahui.

Selain itu, pertimbangkan hal berikut saat Anda mengonfigurasi peran runtime untuk Amazon EMR.

- Amazon EMR mendukung peran runtime di semua iklan. Wilayah AWS
- Langkah-langkah Amazon EMR mendukung pekerjaan Apache Spark dan Apache Hive dengan peran runtime saat Anda menggunakan rilis atau yang lebih baru. `emr-6.7.0`
- SageMaker Studio mendukung kueri Spark, Hive, dan Presto dengan peran runtime saat Anda menggunakan rilis atau yang lebih baru. `emr-6.9.0`
- Kernel notebook berikut dalam SageMaker mendukung peran runtime:
  - DataScience — Kernel Python 3
  - DataScience 2.0 — Kernel Python 3
  - DataScience 3.0 — Kernel Python 3
  - SparkAnalytics 1.0 — SparkMagic dan PySpark kernel
  - SparkAnalytics 2.0 — SparkMagic dan PySpark kernel
  - SparkMagic — PySpark kernel
- Amazon EMR mendukung langkah-langkah yang `RunJobFlow` hanya digunakan pada saat pembuatan cluster. API ini tidak mendukung peran runtime.
- Amazon EMR tidak mendukung peran runtime pada cluster yang Anda konfigurasikan agar sangat tersedia.
- Peran runtime tidak menyediakan dukungan untuk mengontrol akses ke sumber daya on-cluster, seperti HDFS dan HMS.

Konfigurasi peran layanan IAM untuk izin Amazon EMR untuk layanan AWS dan sumber daya

Amazon EMR dan aplikasi seperti Hadoop dan Spark perlu izin untuk mengakses sumber daya AWS lain dan melakukan tindakan ketika mereka dijalankan. Setiap klaster di Amazon EMR harus memiliki peran layanan dan peran untuk profil instans Amazon EC2. Untuk informasi selengkapnya, lihat IAM role dan Menggunakan profil instans di Panduan Pengguna IAM. Kebijakan IAM yang terlampir pada peran ini memberikan izin untuk klaster untuk beroperasi dengan layanan AWS lain atas nama pengguna.

Peran tambahan, peran Auto Scaling, diperlukan jika klaster Anda menggunakan penskalaan otomatis di Amazon EMR. Peran layanan AWS untuk EMR Notebooks diperlukan jika Anda menggunakan EMR Notebooks.

Amazon EMR menyediakan peran default dan kebijakan terkelola default yang menentukan izin untuk setiap peran. Kebijakan terkelola dibuat dan dikelola oleh AWS, sehingga diperbarui secara otomatis jika persyaratan layanan berubah. Lihat AWS kebijakan terkelola di Panduan Pengguna IAM.

Jika Anda membuat sebuah klaster atau notebook untuk pertama kalinya di akun, peran untuk Amazon EMR belum ada. Setelah membuatnya, Anda dapat melihat peran, kebijakan yang dilampirkan padanya, dan izin yang diizinkan atau ditolak oleh kebijakan di konsol IAM (https://console.aws.amazon.com/iam/). Anda dapat menentukan peran default untuk Amazon EMR untuk membuat dan menggunakan, Anda dapat membuat peran Anda sendiri dan menentukan mereka secara individual ketika Anda membuat sebuah klaster untuk menyesuaikan izin, dan Anda dapat menentukan peran default untuk digunakan ketika Anda membuat sebuah klaster menggunakan AWS CLI. Untuk informasi selengkapnya, lihat Kustom IAM role.

Memodifikasi kebijakan berbasis identitas untuk izin dalam melewati peran layanan untuk Amazon EMR

Kebijakan terkelola default izin penuh Amazon EMR menggabungkan konfigurasi `iam:PassRole` keamanan, termasuk yang berikut ini:

- Izin `iam:PassRole` hanya untuk peran Amazon EMR default tertentu.
- `iam:PassedToService`kondisi yang memungkinkan Anda untuk menggunakan kebijakan hanya dengan AWS layanan tertentu, seperti `elasticmapreduce.amazonaws.com` dan`ec2.amazonaws.com`.

Anda dapat melihat versi JSON dari kebijakan AmazonEMR FullAccessPolicy _v2 dan ServicePolicyAmazonEMR _v2 di konsol IAM. Kami menyarankan Anda membuat klaster baru dengan kebijakan terkelola v2.

Ringkasan peran layanan

Tabel berikut mencantumkan peran layanan IAM yang terkait dengan Amazon EMR untuk referensi cepat.

Fungsi	Peran default	Deskripsi	Kebijakan terkelola default
Peran layanan untuk Amazon EMR (peran EMR)	`EMR_DefaultRole_V2`	Mengizinkan Amazon EMR untuk memanggil layanan AWS lain atas nama Anda saat menyediakan sumber daya dan melakukan tindakan tingkat layanan. Peran ini diperlukan untuk semua klaster.	`AmazonEMRServicePolicy_v2` **Important** Peran terkait layanan diperlukan untuk meminta Instans Spot. Jika peran ini tidak ada, peran layanan EMR Amazon harus memiliki izin untuk membuatnya atau kesalahan izin terjadi. Jika Anda berencana untuk meminta

Fungsi	Peran default	Deskripsi	Kebijakan terkelola default
			Instans Spot, Anda harus memperbarui kebijakan ini untuk menyertakan pernyataan yang memungkinkan pembuatan peran terkait layanan ini. Untuk informasi selengkapnya, lihat Peran layanan untuk Amazon EMR (peran EMR) dan peran tertaut layanan untuk permintaan Instans Spot di Panduan Pengguna Amazon EC2 untuk Instans Linux.

Fungsi	Peran default	Deskripsi	Kebijakan terkelola default
Peran layanan untuk instans EC2 klaster (profil instans EC2)	`EMR_EC2_DefaultRole`	Proses aplikasi yang berjalan di atas ekosistem Hadoop pada instans klaster menggunakan peran ini ketika mereka memanggil layanan AWS lainnya. Untuk mengakses data di Amazon S3 menggunakan EMRFS, Anda dapat menentukan peran yang berbeda untuk diasumsikan berdasarkan lokasi data di Amazon S3. Misalnya, beberapa tim dapat mengakses "akun penyimpanan" data Amazon S3 tunggal. Untuk informasi selengkapnya, lihat Konfigurasi IAM role untuk permintaan EMRFS ke Amazon S3. Peran ini diperlukan untuk semua klaster.	`AmazonElasticMapReduceforEC2Role`. Untuk informasi selengkapnya, lihat Peran layanan untuk instans EC2 klaster (profil instans EC2).

Fungsi	Peran default	Deskripsi	Kebijakan terkelola default
Peran layanan untuk penskalaan otomatis di Amazon EMR (peran Auto Scaling)	`EMR_AutoScaling_DefaultRole`	Mengizinkan tindakan tambahan untuk lingkungan penskalaan dinamis. Diperlukan hanya untuk klaster yang menggunakan penskalaan otomatis di Amazon EMR. Untuk informasi selengkapnya, lihat Menggunakan penskalaan otomatis dengan kebijakan kustom untuk grup instans .	`AmazonElasticMapReduceforAutoScalingRole` . Untuk informasi selengkapnya, lihat Peran layanan untuk penskalaan otomatis di Amazon EMR (peran Auto Scaling).

Fungsi	Peran default	Deskripsi	Kebijakan terkelola default
Peran layanan untuk EMR Notebooks	`EMR_Notebooks_DefaultRole`	Memberikan izin yang dibutuhkan EMR Notebooks untuk mengakses AWS sumber daya dan melakukan tindakan. Diperlukan hanya jika EMR Notebooks digunakan.	`AmazonElasticMapReduceEditorsRole` . Untuk informasi selengkapnya, lihat Peran layanan untuk EMR Notebooks. `S3FullAccessPolicy` juga dilampirkan secara default. Berikut adalah isi dari kebijakan ini. `{ "Version": "2012-10-17", "Statement": [ { "Effect": "Allow", "Action": "s3:*", "Resource": "*" } ] }`

Fungsi	Peran default	Deskripsi	Kebijakan terkelola default
Peran Terkait Layanan	`AWSServiceRoleForEMRCleanup`	Amazon EMR secara otomatis menciptakan peran tertaut layanan. Jika layanan untuk Amazon EMR telah kehilangan kemampuan untuk membersihkan sumber daya Amazon EC2, Amazon EMR dapat menggunakan peran ini untuk membersihkan. Jika klaster menggunakan Instans Spot, kebijakan izin yang dilampirkan ke Peran layanan untuk Amazon EMR (peran EMR) harus mengizinkan pembuatan peran tertaut layanan. Untuk informasi selengkapnya, lihat Izin peran tertaut layanan untuk Amazon EMR.	`AmazonEMRCleanupPolicy`

Topik

- Peran layanan IAM yang digunakan oleh Amazon EMR
- Kustom IAM role
- Konfigurasi IAM role untuk permintaan EMRFS ke Amazon S3

- Gunakan kebijakan berbasis sumber daya untuk akses Amazon EMR ke AWS Katalog Data Glue
- Menggunakan IAM role dengan aplikasi yang memanggil layanan AWS secara langsung
- Mengizinkan pengguna dan grup untuk membuat dan memodifikasi peran

Peran layanan IAM yang digunakan oleh Amazon EMR

Amazon EMR menggunakan peran layanan IAM untuk melakukan tindakan atas nama Anda ketika menyediakan sumber daya klaster, menjalankan aplikasi, menskalakan sumber daya secara dinamis, dan menciptakan dan menjalankan EMR Notebooks. Amazon EMR menggunakan peran berikut ketika berinteraksi dengan layanan AWS lain. Setiap peran memiliki fungsi yang unik di Amazon EMR. Topik di bagian ini menjelaskan fungsi peran dan menyediakan peran default dan kebijakan izin untuk setiap peran.

Jika Anda memiliki kode aplikasi di klaster Anda yang memanggil AWS layanan langsung, Anda mungkin perlu menggunakan SDK untuk menentukan peran. Untuk informasi selengkapnya, lihat Menggunakan IAM role dengan aplikasi yang memanggil layanan AWS secara langsung.

Topik

- Peran layanan untuk Amazon EMR (peran EMR)
- Peran layanan untuk instans EC2 klaster (profil instans EC2)
- Peran layanan untuk penskalaan otomatis di Amazon EMR (peran Auto Scaling)
- Peran layanan untuk EMR Notebooks
- Menggunakan peran yang berkaitan dengan layanan untuk Amazon EMR

Peran layanan untuk Amazon EMR (peran EMR)

Peran Amazon EMR mendefinisikan tindakan yang diizinkan untuk Amazon EMR saat menyediakan sumber daya dan melakukan tugas tingkat layanan yang tidak dilakukan dalam konteks instans Amazon EC2 yang berjalan dalam klaster. Misalnya, peran layanan yang digunakan untuk menyediakan instans EC2 ketika sebuah klaster meluncur.

- Nama peran default adalah `EMR_DefaultRole_V2`.
- Amazon EMR melingkupi kebijakn terkelola default yang terlampir pada `EMR_DefaultRole_V2` adalah `AmazonEMRServicePolicy_v2`. Kebijakan v2 ini menggantikan kebijakan terkelola default yang tidak digunakan lagi. `AmazonElasticMapReduceRole`

`AmazonEMRServicePolicy_v2`bergantung pada akses terbatas ke sumber daya yang disediakan atau digunakan Amazon EMR. Bila menggunakan kebijakan ini, Anda harus melewati tanda pengguna `for-use-with-amazon-emr-managed-policies = true` saat menyediakan klaster. Amazon EMR akan secara otomatis menyebarkan tag tersebut. Selain itu, Anda mungkin perlu secara manual menambahkan tanda pengguna untuk tipe sumber daya tertentu, seperti grup keamanan EC2 yang tidak dibuat oleh Amazon EMR. Lihat Penandaan sumber daya untuk menggunakan kebijakan terkelola.

> **Important**
>
> Amazon EMR menggunakan peran layanan EMR Amazon ini dan `AWSServiceRoleForEMRCleanup` peran untuk membersihkan sumber daya cluster di akun Anda yang tidak lagi Anda gunakan, seperti instans Amazon EC2. Anda harus menyertakan tindakan agar kebijakan peran menghapus atau menghentikan sumber daya. Jika tidak, Amazon EMR tidak dapat melakukan tindakan pembersihan ini, dan Anda mungkin dikenakan biaya untuk sumber daya yang tidak digunakan yang tetap ada di klaster.

Berikut ini menunjukkan isi arus `AmazonEMRServicePolicy_v2` kebijakan. Anda juga dapat melihat konten kebijakan `AmazonEMRServicePolicy_v2`terkelola saat ini di konsol IAM.

```
{
 "Version": "2012-10-17",
 "Statement": [
  {
   "Sid": "CreateInTaggedNetwork",
   "Effect": "Allow",
   "Action": [
    "ec2:CreateNetworkInterface",
    "ec2:RunInstances",
    "ec2:CreateFleet",
    "ec2:CreateLaunchTemplate",
    "ec2:CreateLaunchTemplateVersion"
   ],
   "Resource": [
    "arn:aws:ec2:*:*:subnet/*",
    "arn:aws:ec2:*:*:security-group/*"
   ],
   "Condition": {
    "StringEquals": {
```

```
      "aws:ResourceTag/for-use-with-amazon-emr-managed-policies": "true"
     }
    }
   },
   {
    "Sid": "CreateWithEMRTaggedLaunchTemplate",
    "Effect": "Allow",
    "Action": [
     "ec2:CreateFleet",
     "ec2:RunInstances",
     "ec2:CreateLaunchTemplateVersion"
    ],
    "Resource": "arn:aws:ec2:*:*:launch-template/*",
    "Condition": {
     "StringEquals": {
      "aws:ResourceTag/for-use-with-amazon-emr-managed-policies": "true"
     }
    }
   },
   {
    "Sid": "CreateEMRTaggedLaunchTemplate",
    "Effect": "Allow",
    "Action": "ec2:CreateLaunchTemplate",
    "Resource": "arn:aws:ec2:*:*:launch-template/*",
    "Condition": {
     "StringEquals": {
      "aws:RequestTag/for-use-with-amazon-emr-managed-policies": "true"
     }
    }
   },
   {
    "Sid": "CreateEMRTaggedInstancesAndVolumes",
    "Effect": "Allow",
    "Action": [
     "ec2:RunInstances",
     "ec2:CreateFleet"
    ],
    "Resource": [
     "arn:aws:ec2:*:*:instance/*",
     "arn:aws:ec2:*:*:volume/*"
    ],
    "Condition": {
     "StringEquals": {
      "aws:RequestTag/for-use-with-amazon-emr-managed-policies": "true"
```

```
    }
   }
  },
  {
   "Sid": "ResourcesToLaunchEC2",
   "Effect": "Allow",
   "Action": [
    "ec2:RunInstances",
    "ec2:CreateFleet",
    "ec2:CreateLaunchTemplate",
    "ec2:CreateLaunchTemplateVersion"
   ],
   "Resource": [
    "arn:aws:ec2:*:*:network-interface/*",
    "arn:aws:ec2:*::image/ami-*",
    "arn:aws:ec2:*:*:key-pair/*",
    "arn:aws:ec2:*:*:capacity-reservation/*",
    "arn:aws:ec2:*:*:placement-group/pg-*",
    "arn:aws:ec2:*:*:fleet/*",
    "arn:aws:ec2:*:*:dedicated-host/*",
    "arn:aws:resource-groups:*:*:group/*"
   ]
  },
  {
   "Sid": "ManageEMRTaggedResources",
   "Effect": "Allow",
   "Action": [
    "ec2:CreateLaunchTemplateVersion",
    "ec2:DeleteLaunchTemplate",
    "ec2:DeleteNetworkInterface",
    "ec2:ModifyInstanceAttribute",
    "ec2:TerminateInstances"
   ],
   "Resource": "*",
   "Condition": {
    "StringEquals": {
     "aws:ResourceTag/for-use-with-amazon-emr-managed-policies": "true"
    }
   }
  },
  {
   "Sid": "ManageTagsOnEMRTaggedResources",
   "Effect": "Allow",
   "Action": [
```

```
    "ec2:CreateTags",
    "ec2:DeleteTags"
   ],
   "Resource": [
    "arn:aws:ec2:*:*:instance/*",
    "arn:aws:ec2:*:*:volume/*",
    "arn:aws:ec2:*:*:network-interface/*",
    "arn:aws:ec2:*:*:launch-template/*"
   ],
   "Condition": {
    "StringEquals": {
     "aws:ResourceTag/for-use-with-amazon-emr-managed-policies": "true"
    }
   }
  },
  {
   "Sid": "CreateNetworkInterfaceNeededForPrivateSubnet",
   "Effect": "Allow",
   "Action": [
    "ec2:CreateNetworkInterface"
   ],
   "Resource": [
    "arn:aws:ec2:*:*:network-interface/*"
   ],
   "Condition": {
    "StringEquals": {
     "aws:RequestTag/for-use-with-amazon-emr-managed-policies": "true"
    }
   }
  },
  {
   "Sid": "TagOnCreateTaggedEMRResources",
   "Effect": "Allow",
   "Action": [
    "ec2:CreateTags"
   ],
   "Resource": [
    "arn:aws:ec2:*:*:network-interface/*",
    "arn:aws:ec2:*:*:instance/*",
    "arn:aws:ec2:*:*:volume/*",
    "arn:aws:ec2:*:*:launch-template/*"
   ],
   "Condition": {
    "StringEquals": {
```

```
     "ec2:CreateAction": [
      "RunInstances",
      "CreateFleet",
      "CreateLaunchTemplate",
      "CreateNetworkInterface"
     ]
    }
   }
  },
  {
   "Sid": "TagPlacementGroups",
   "Effect": "Allow",
   "Action": [
    "ec2:CreateTags",
    "ec2:DeleteTags"
   ],
   "Resource": [
    "arn:aws:ec2:*:*:placement-group/pg-*"
   ]
  },
  {
   "Sid": "ListActionsForEC2Resources",
   "Effect": "Allow",
   "Action": [
    "ec2:DescribeAccountAttributes",
    "ec2:DescribeCapacityReservations",
    "ec2:DescribeDhcpOptions",
    "ec2:DescribeImages",
    "ec2:DescribeInstances",
    "ec2:DescribeLaunchTemplates",
    "ec2:DescribeNetworkAcls",
    "ec2:DescribeNetworkInterfaces",
    "ec2:DescribePlacementGroups",
    "ec2:DescribeRouteTables",
    "ec2:DescribeSecurityGroups",
    "ec2:DescribeSubnets",
    "ec2:DescribeVolumes",
    "ec2:DescribeVolumeStatus",
    "ec2:DescribeVpcAttribute",
    "ec2:DescribeVpcEndpoints",
    "ec2:DescribeVpcs"
   ],
   "Resource": "*"
  },
```

```
  {
   "Sid": "CreateDefaultSecurityGroupWithEMRTags",
   "Effect": "Allow",
   "Action": [
    "ec2:CreateSecurityGroup"
   ],
   "Resource": [
    "arn:aws:ec2:*:*:security-group/*"
   ],
   "Condition": {
    "StringEquals": {
     "aws:RequestTag/for-use-with-amazon-emr-managed-policies": "true"
    }
   }
  },
  {
   "Sid": "CreateDefaultSecurityGroupInVPCWithEMRTags",
   "Effect": "Allow",
   "Action": [
    "ec2:CreateSecurityGroup"
   ],
   "Resource": [
    "arn:aws:ec2:*:*:vpc/*"
   ],
   "Condition": {
    "StringEquals": {
     "aws:ResourceTag/for-use-with-amazon-emr-managed-policies": "true"
    }
   }
  },
  {
   "Sid": "TagOnCreateDefaultSecurityGroupWithEMRTags",
   "Effect": "Allow",
   "Action": [
    "ec2:CreateTags"
   ],
   "Resource": "arn:aws:ec2:*:*:security-group/*",
   "Condition": {
    "StringEquals": {
     "aws:RequestTag/for-use-with-amazon-emr-managed-policies": "true",
     "ec2:CreateAction": "CreateSecurityGroup"
    }
   }
  },
```

```
{
 "Sid": "ManageSecurityGroups",
 "Effect": "Allow",
 "Action": [
  "ec2:AuthorizeSecurityGroupEgress",
  "ec2:AuthorizeSecurityGroupIngress",
  "ec2:RevokeSecurityGroupEgress",
  "ec2:RevokeSecurityGroupIngress"
 ],
 "Resource": "*",
 "Condition": {
  "StringEquals": {
   "aws:ResourceTag/for-use-with-amazon-emr-managed-policies": "true"
  }
 }
},
{
 "Sid": "CreateEMRPlacementGroups",
 "Effect": "Allow",
 "Action": [
  "ec2:CreatePlacementGroup"
 ],
 "Resource": "arn:aws:ec2:*:*:placement-group/pg-*"
},
{
 "Sid": "DeletePlacementGroups",
 "Effect": "Allow",
 "Action": [
  "ec2:DeletePlacementGroup"
 ],
 "Resource": "*"
},
{
 "Sid": "AutoScaling",
 "Effect": "Allow",
 "Action": [
  "application-autoscaling:DeleteScalingPolicy",
  "application-autoscaling:DeregisterScalableTarget",
  "application-autoscaling:DescribeScalableTargets",
  "application-autoscaling:DescribeScalingPolicies",
  "application-autoscaling:PutScalingPolicy",
  "application-autoscaling:RegisterScalableTarget"
 ],
 "Resource": "*"
```

```
  },
  {
   "Sid": "ResourceGroupsForCapacityReservations",
   "Effect": "Allow",
   "Action": [
    "resource-groups:ListGroupResources"
   ],
   "Resource": "*"
  },
  {
   "Sid": "AutoScalingCloudWatch",
   "Effect": "Allow",
   "Action": [
    "cloudwatch:PutMetricAlarm",
    "cloudwatch:DeleteAlarms",
    "cloudwatch:DescribeAlarms"
   ],
   "Resource": "arn:aws:cloudwatch:*:*:alarm:*_EMR_Auto_Scaling"
  },
  {
   "Sid": "PassRoleForAutoScaling",
   "Effect": "Allow",
   "Action": "iam:PassRole",
   "Resource": "arn:aws:iam::*:role/EMR_AutoScaling_DefaultRole",
   "Condition": {
    "StringLike": {
     "iam:PassedToService": "application-autoscaling.amazonaws.com*"
    }
   }
  },
  {
   "Sid": "PassRoleForEC2",
   "Effect": "Allow",
   "Action": "iam:PassRole",
   "Resource": "arn:aws:iam::*:role/EMR_EC2_DefaultRole",
   "Condition": {
    "StringLike": {
     "iam:PassedToService": "ec2.amazonaws.com*"
    }
   }
  }
 ]
}
```

Peran layanan Anda harus menggunakan kebijakan kepercayaan berikut.

> **Important**
>
> Kebijakan kepercayaan berikut mencakup `aws:SourceArn` dan kunci kondisi `aws:SourceAccount` global, yang membatasi izin yang Anda berikan EMR Amazon ke sumber daya tertentu di akun Anda. Menggunakannya dapat melindungi Anda dari masalah wakil yang membingungkan.

```
{
    "Version": "2008-10-17",
    "Statement": [
        {
            "Sid": "",
            "Effect": "Allow",
            "Principal": {
                "Service": "elasticmapreduce.amazonaws.com"
            },
            "Action": "sts:AssumeRole",
            "Condition": {
                "StringEquals": {
                    "aws:SourceAccount": "<account-id>"
                },
                "ArnLike": {
                    "aws:SourceArn": "arn:aws:elasticmapreduce:<region>:<account-id>:*"
                }
            }
        }
    ]
}
```

Peran layanan untuk instans EC2 klaster (profil instans EC2)

Peran layanan untuk instans EC2 klaster (juga disebut profil instans EC2 untuk Amazon EMR) adalah tipe khusus dari peran layanan yang ditugaskan untuk setiap instans EC2 di sebuah klaster Amazon EMR ketika instans meluncur. Proses aplikasi yang berjalan di atas ekosistem Hadoop menganggap peran ini untuk izin untuk berinteraksi dengan layanan AWS lain.

Untuk informasi lebih lanjut tentang peran layanan untuk instans EC2, lihat Menggunakan IAM role untuk memberikan izin pada aplikasi yang berjalan di instans Amazon EC2 di Panduan Pengguna IAM.

> **Important**
>
> Peran layanan default untuk instans EC2 klaster dan yang terkait AWS kebijakan terkelola default, `AmazonElasticMapReduceforEC2Role` tidak lagi digunakan, tanpa penggantian AWS kebijakan terkelola yang disediakan. Anda harus membuat dan menentukan profil instans untuk mengganti peran dan kebijakan default yang tidak lagi digunakan.

Peran default dan kebijakan terkelola

- Nama peran default adalah `EMR_EC2_DefaultRole`.
- Kebijakan terkelola `EMR_EC2_DefaultRole` default`AmazonElasticMapReduceforEC2Role`,, mendekati akhir dukungan. Alih-alih menggunakan kebijakan terkelola default untuk profil instans EC2, terapkan kebijakan berbasis sumber daya ke bucket S3 dan sumber daya lain yang dibutuhkan Amazon EMR, atau gunakan kebijakan yang dikelola pelanggan Anda sendiri dengan peran IAM sebagai profil instans. Untuk informasi selengkapnya, lihat Membuat peran layanan untuk instans EC2 klaster dengan izin hak istimewa paling sedikit.

Berikut ini menunjukkan isi dari versi 3 dari `AmazonElasticMapReduceforEC2Role`.

```
{
    "Version": "2012-10-17",
    "Statement": [
        {
            "Effect": "Allow",
            "Resource": "*",
            "Action": [
                "cloudwatch:*",
                "dynamodb:*",
                "ec2:Describe*",
                "elasticmapreduce:Describe*",
                "elasticmapreduce:ListBootstrapActions",
                "elasticmapreduce:ListClusters",
                "elasticmapreduce:ListInstanceGroups",
                "elasticmapreduce:ListInstances",
                "elasticmapreduce:ListSteps",
```

```
                "kinesis:CreateStream",
                "kinesis:DeleteStream",
                "kinesis:DescribeStream",
                "kinesis:GetRecords",
                "kinesis:GetShardIterator",
                "kinesis:MergeShards",
                "kinesis:PutRecord",
                "kinesis:SplitShard",
                "rds:Describe*",
                "s3:*",
                "sdb:*",
                "sns:*",
                "sqs:*",
                "glue:CreateDatabase",
                "glue:UpdateDatabase",
                "glue:DeleteDatabase",
                "glue:GetDatabase",
                "glue:GetDatabases",
                "glue:CreateTable",
                "glue:UpdateTable",
                "glue:DeleteTable",
                "glue:GetTable",
                "glue:GetTables",
                "glue:GetTableVersions",
                "glue:CreatePartition",
                "glue:BatchCreatePartition",
                "glue:UpdatePartition",
                "glue:DeletePartition",
                "glue:BatchDeletePartition",
                "glue:GetPartition",
                "glue:GetPartitions",
                "glue:BatchGetPartition",
                "glue:CreateUserDefinedFunction",
                "glue:UpdateUserDefinedFunction",
                "glue:DeleteUserDefinedFunction",
                "glue:GetUserDefinedFunction",
                "glue:GetUserDefinedFunctions"
            ]
        }
    ]
}
```

Peran layanan Anda harus menggunakan kebijakan kepercayaan berikut.

```
{
    "Version": "2008-10-17",
    "Statement": [
        {
            "Sid": "",
            "Effect": "Allow",
            "Principal": {
                "Service": "ec2.amazonaws.com"
            },
            "Action": "sts:AssumeRole"
        }
    ]
}
```

Membuat peran layanan untuk instans EC2 klaster dengan izin hak istimewa paling sedikit

Sebagai praktik terbaik, kami sangat merekomendasikan agar Anda membuat peran layanan untuk instans EC2 klaster dan kebijakan izin yang memiliki izin minimum untuk layanan AWS lain yang dibutuhkan oleh aplikasi Anda.

Kebijakan terkelola default, `AmazonElasticMapReduceforEC2Role`, menyediakan izin yang membuatnya mudah untuk meluncurkan klaster awal. Namun, `AmazonElasticMapReduceforEC2Role` tidak lagi digunakan dan Amazon EMR tidak akan menyediakan pengganti AWS kebijakan default terkelola untuk peran yang tidak lagi digunakan. Untuk meluncurkan klaster awal, Anda perlu menyediakan pelanggan terkelola berbasis sumber daya atau kebijakan berbasis ID.

Pernyataan kebijakan berikut memberikan contoh izin yang diperlukan untuk fitur yang berbeda dari Amazon EMR. Kami merekomendasikan Anda menggunakan izin ini untuk membuat kebijakan izin yang membatasi akses ke fitur dan sumber daya yang hanya diperlukan klaster Anda. Semua contoh pernyataan kebijakan menggunakan *`us-west-2`*Region dan ID AWS *`123456789012`*akun fiksi. Ganti ini agar sesuai untuk klaster Anda.

Untuk informasi selengkapnya tentang pembuatan dan penentuan peran kustom, lihat Kustom IAM role.

> **Note**
>
> Jika Anda membuat peran EMR kustom untuk EC2, ikuti alur kerja basic, yang secara otomatis membuat profil instans dengan nama yang sama. Amazon EC2 mengizinkan Anda untuk membuat profil instans dan peran dengan nama yang berbeda, tetapi Amazon EMR

tidak support konfigurasi ini, dan itu menghasilkan kesalahan "profil instans tidak valid" ketika Anda membuat klaster.

Membaca dan menulis data ke Amazon S3 menggunakan EMRFS

Ketika aplikasi yang berjalan di data referensi klaster Amazon EMR menggunakan format `s3://mydata`, Amazon EMR menggunakan profil instans EC2 untuk membuat permintaan. Klaster biasanya membaca dan menulis data ke Amazon S3 dengan cara ini, dan Amazon EMR menggunakan izin yang terlampir pada peran layanan untuk instans EC2 klaster secara default. Untuk informasi selengkapnya, lihat Konfigurasi IAM role untuk permintaan EMRFS ke Amazon S3.

Karena IAM role untuk EMRFS akan mundur ke izin yang terlampir pada peran layanan untuk instans EC2 klaster, sebagai praktik terbaik, kami rekomendasikan Anda menggunakan IAM role untuk EMRFS, dan membatasi izin EMRFS dan Amazon S3 yang terlampir pada peran layanan untuk instans EC2 klaster.

Sampel pernyataan di bawah ini menunjukkan izin yang diperlukan EMRFS untuk membuat permintaan ke Amazon S3.

- *my-data-bucket-in-s3-for-emrfs-reads-and-writes*menentukan bucket di Amazon S3 tempat cluster membaca dan menulis data dan semua sub-folder yang digunakan. */** Menambahkan hanya bucket dan folder yang dibutuhkan aplikasi Anda.
- Pernyataan kebijakan yang mengizinkan dynamodb tindakan hanya diperlukan jika tampilan konsisten EMRFS diaktifkan. *EmrFSMetadata*menentukan folder default untuk tampilan konsisten EMRFS.

```
{
    "Version": "2012-10-17",
    "Statement": [
        {
            "Effect": "Allow",
            "Action": [
                "s3:AbortMultipartUpload",
                "s3:CreateBucket",
                "s3:DeleteObject",
                "s3:GetBucketVersioning",
                "s3:GetObject",
                "s3:GetObjectTagging",
```

```
                "s3:GetObjectVersion",
                "s3:ListBucket",
                "s3:ListBucketMultipartUploads",
                "s3:ListBucketVersions",
                "s3:ListMultipartUploadParts",
                "s3:PutBucketVersioning",
                "s3:PutObject",
                "s3:PutObjectTagging"
            ],
            "Resource": [
                "arn:aws:s3:::my-data-bucket-in-s3-for-emrfs-reads-and-writes",
                "arn:aws:s3:::my-data-bucket-in-s3-for-emrfs-reads-and-writes/*"
            ]
        },
        {
            "Effect": "Allow",
            "Action": [
                "dynamodb:CreateTable",
                "dynamodb:BatchGetItem",
                "dynamodb:BatchWriteItem",
                "dynamodb:PutItem",
                "dynamodb:DescribeTable",
                "dynamodb:DeleteItem",
                "dynamodb:GetItem",
                "dynamodb:Scan",
                "dynamodb:Query",
                "dynamodb:UpdateItem",
                "dynamodb:DeleteTable",
                "dynamodb:UpdateTable"
            ],
            "Resource": "arn:aws:dynamodb:us-west-2:123456789012:table/EmrFSMetadata"
        },
        {
            "Effect": "Allow",
            "Action": [
                "cloudwatch:PutMetricData",
                "dynamodb:ListTables",
                "s3:ListBucket"
            ],
            "Resource": "*"
        },
        {
            "Effect": "Allow",
            "Action": [
```

```
                "sqs:GetQueueUrl",
                "sqs:ReceiveMessage",
                "sqs:DeleteQueue",
                "sqs:SendMessage",
                "sqs:CreateQueue"
            ],
            "Resource": "arn:aws:sqs:us-west-2:123456789012:EMRFS-Inconsistency-*"
        }
    ]
}
```

Mengarsipkan log file ke Amazon S3

Pernyataan kebijakan berikut mengizinkan klaster Amazon EMR untuk log file arsip ke lokasi Amazon S3 yang ditentukan. Dalam contoh di bawah ini, ketika cluster *s3://MyLoggingBucket/MyEMRClusterLogs*dibuat, ditentukan menggunakan lokasi folder Log S3 di konsol, menggunakan `--log-uri` opsi dariAWS CLI, atau menggunakan `LogUri` parameter dalam `RunJobFlow` perintah. Untuk informasi selengkapnya, lihat Arsipkan berkas log ke Amazon S3.

```
{
    "Version": "2012-10-17",
    "Statement": [
        {
            "Effect": "Allow",
            "Action": "s3:PutObject",
            "Resource": "arn:aws:s3:::MyLoggingBucket/MyEMRClusterLogs/*"
        }
    ]
}
```

Menggunakan alat debugging

Pernyataan kebijakan berikut mengizinkan tindakan yang diperlukan jika Anda mengaktifkan alat debugging Amazon EMR. Mengarsipkan log file ke Amazon S3, dan izin terkait yang ditunjukkan contoh di atas, diperlukan untuk debugging. Untuk informasi selengkapnya, lihat Aktifkan alat debugging.

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

```
        {
            "Effect": "Allow",
            "Action": [
                "sqs:GetQueueUrl",
                "sqs:SendMessage"
            ],
            "Resource": "arn:aws:sqs:us-west-2:123456789012:AWS-ElasticMapReduce-*"
        }
    ]
}
```

Menggunakan AWS Katalog Data Glue

Pernyataan kebijakan berikut mengizinkan tindakan yang diperlukan jika Anda menggunakan AWS Katalog Data Glue sebagai metastore untuk aplikasi. Untuk informasi selengkapnya, lihat Menggunakan AWS Katalog Data Glue sebagai metastore untuk Spark SQL, Menggunakan AWS Katalog Data Glue sebagai metastore untuk Hive, dan Menggunakan Presto dengan AWS Katalog Data Glue di Panduan Amazon EMR rilis.

```
{
    "Version": "2012-10-17",
    "Statement": [
        {
            "Effect": "Allow",
            "Action": [
                "glue:CreateDatabase",
                "glue:UpdateDatabase",
                "glue:DeleteDatabase",
                "glue:GetDatabase",
                "glue:GetDatabases",
                "glue:CreateTable",
                "glue:UpdateTable",
                "glue:DeleteTable",
                "glue:GetTable",
                "glue:GetTables",
                "glue:GetTableVersions",
                "glue:CreatePartition",
                "glue:BatchCreatePartition",
                "glue:UpdatePartition",
                "glue:DeletePartition",
                "glue:BatchDeletePartition",
                "glue:GetPartition",
                "glue:GetPartitions",
```

```
                "glue:BatchGetPartition",
                "glue:CreateUserDefinedFunction",
                "glue:UpdateUserDefinedFunction",
                "glue:DeleteUserDefinedFunction",
                "glue:GetUserDefinedFunction",
                "glue:GetUserDefinedFunctions"
            ],
            "Resource": "*",
        }
    ]
}
```

Peran layanan untuk penskalaan otomatis di Amazon EMR (peran Auto Scaling)

Peran Auto Scaling untuk Amazon EMR menjalankan fungsi yang sama seperti peran layanan, tetapi memungkinkan tindakan tambahan untuk lingkungan penskalaan dinamis.

- Nama peran default adalah `EMR_AutoScaling_DefaultRole`.
- Kebijakan terkelola default yang terlampir pada `EMR_AutoScaling_DefaultRole` adalah `AmazonElasticMapReduceforAutoScalingRole`.

Isi dari versi 1 dari `AmazonElasticMapReduceforAutoScalingRole` ditunjukkan di bawah ini.

```
{
    "Version": "2012-10-17",
    "Statement": [
        {
            "Action": [
                "cloudwatch:DescribeAlarms",
                "elasticmapreduce:ListInstanceGroups",
                "elasticmapreduce:ModifyInstanceGroups"
            ],
            "Effect": "Allow",
            "Resource": "*"
        }
    ]
}
```

Peran layanan Anda harus menggunakan kebijakan kepercayaan berikut.

> **Important**
>
> Kebijakan kepercayaan berikut mencakup `aws:SourceArn`dan kunci kondisi `aws:SourceAccount`global, yang membatasi izin yang Anda berikan EMR Amazon ke sumber daya tertentu di akun Anda. Menggunakannya dapat melindungi Anda dari masalah wakil yang membingungkan.

```
{
    "Version": "2012-10-17",
    "Statement": [
        {
            "Effect": "Allow",
            "Principal": {
                "Service":  [
                    "elasticmapreduce.amazonaws.com",
                    "application-autoscaling.amazonaws.com"
                ]
            },
            "Action": "sts:AssumeRole",
            "Condition": {
                "StringEquals": {
                    "aws:SourceAccount": "<account-id>"
                },
                "ArnLike": {
                    "aws:SourceArn": "arn:aws:elasticmapreduce:<region>:<account-id>:*"
                }
            }
        }
    ]
}
```

Peran layanan untuk EMR Notebooks

Setiap EMR Notebooks memerlukan izin untuk mengakses AWS sumber daya dan melakukan tindakan. Kebijakan IAM yang dilampirkan pada peran layanan ini memberikan izin bagi notebook untuk beroperasi dengan layanan AWS lain. Saat Anda membuat notebook menggunakan AWS Management Console, Anda menentukan AWS peran layanan. Anda dapat menggunakan peran default, `EMR_Notebooks_DefaultRole`, atau tentukan peran yang Anda buat. Jika notebook belum dibuat sebelumnya, Anda dapat memilih untuk membuat peran default.

- Nama peran default adalah `EMR_Notebooks_DefaultRole`.
- Kebijakan terkelola default yang dilampirkan `EMR_Notebooks_DefaultRole` adalah `AmazonElasticMapReduceEditorsRole` dan`S3FullAccessPolicy`.

Peran layanan Anda harus menggunakan kebijakan kepercayaan berikut.

> **Important**
>
> Kebijakan kepercayaan berikut mencakup `aws:SourceArn`dan kunci kondisi `aws:SourceAccount`global, yang membatasi izin yang Anda berikan EMR Amazon ke sumber daya tertentu di akun Anda. Menggunakannya dapat melindungi Anda dari masalah wakil yang membingungkan.

```
{
    "Version": "2012-10-17",
    "Statement": [
        {
            "Effect": "Allow",
            "Principal": {
                "Service": "elasticmapreduce.amazonaws.com"
            },
            "Action": "sts:AssumeRole",
            "Condition": {
                "StringEquals": {
                    "aws:SourceAccount": "<account-id>"
                },
                "ArnLike": {
                    "aws:SourceArn": "arn:aws:elasticmapreduce:<region>:<account-id>:*"
                }
            }
        }
    ]
}
```

Isi dari versi 1 `AmazonElasticMapReduceEditorsRole` adalah sebagai berikut.

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

```
        {
            "Effect": "Allow",
            "Action": [
                "ec2:AuthorizeSecurityGroupEgress",
                "ec2:AuthorizeSecurityGroupIngress",
                "ec2:CreateSecurityGroup",
                "ec2:DescribeSecurityGroups",
                "ec2:RevokeSecurityGroupEgress",
                "ec2:CreateNetworkInterface",
                "ec2:CreateNetworkInterfacePermission",
                "ec2:DeleteNetworkInterface",
                "ec2:DeleteNetworkInterfacePermission",
                "ec2:DescribeNetworkInterfaces",
                "ec2:ModifyNetworkInterfaceAttribute",
                "ec2:DescribeTags",
                "ec2:DescribeInstances",
                "ec2:DescribeSubnets",
                "ec2:DescribeVpcs",
                "elasticmapreduce:ListInstances",
                "elasticmapreduce:DescribeCluster",
                "elasticmapreduce:ListSteps"
            ],
            "Resource": "*"
        },
        {
            "Effect": "Allow",
            "Action": "ec2:CreateTags",
            "Resource": "arn:aws:ec2:*:*:network-interface/*",
            "Condition": {
                "ForAllValues:StringEquals": {
                    "aws:TagKeys": [
                        "aws:elasticmapreduce:editor-id",
                        "aws:elasticmapreduce:job-flow-id"
                    ]
                }
            }
        }
    ]
}
```

Berikut ini adalah isi dari`S3FullAccessPolicy`. `S3FullAccessPolicy`Ini memungkinkan peran layanan Anda untuk EMR Notebooks untuk melakukan semua tindakan Amazon S3 pada objek di

Anda. Akun AWS Saat Anda membuat peran layanan kustom untuk EMR Notebooks, Anda harus memberikan izin Amazon S3 peran layanan Anda.

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

Anda dapat mencatat akses baca dan tulis untuk peran layanan Anda ke lokasi Amazon S3 tempat Anda ingin menyimpan file notebook. Gunakan set minimum izin Amazon S3 berikut.

```
"s3:PutObject",
"s3:GetObject",
"s3:GetEncryptionConfiguration",
"s3:ListBucket",
"s3:DeleteObject"
```

Jika bucket Amazon S3 Anda dienkripsi, Anda harus menyertakan izin berikut untuk. AWS Key Management Service

```
"kms:Decrypt",
"kms:GenerateDataKey",
"kms:ReEncryptFrom",
"kms:ReEncryptTo",
"kms:DescribeKey"
```

Saat Anda menautkan repositori Git ke buku catatan Anda dan perlu membuat rahasia untuk repositori, Anda harus menambahkan `secretsmanager:GetSecretValue` izin dalam kebijakan IAM yang dilampirkan ke peran layanan untuk notebook Amazon EMR. Kebijakan contoh ditunjukkan di bawah ini:

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

```
            "Sid": "VisualEditor0",
            "Effect": "Allow",
            "Action": "secretsmanager:GetSecretValue",
            "Resource": "*"
        }
    ]
}
```

Izin peran layanan EMR Notebooks

Tabel ini mencantumkan tindakan yang dilakukan EMR Notebooks menggunakan peran layanan, bersama dengan izin yang diperlukan untuk setiap tindakan.

Tindakan	Izin
Buat saluran jaringan aman antara notebook dan kluster EMR Amazon, dan lakukan tindakan pembersihan yang diperlukan.	`"ec2:CreateNetworkInterface", "ec2:CreateNetworkInterfacePermission", "ec2:DeleteNetworkInterface", "ec2:DeleteNetworkInterfacePermission", "ec2:DescribeNetworkInterfaces", "ec2:ModifyNetworkInterfaceAttribute", "ec2:AuthorizeSecurityGroupEgress", "ec2:AuthorizeSecurityGroupIngress", "ec2:CreateSecurityGroup", "ec2:DescribeSecurityGroups", "ec2:RevokeSecurityGroupEgress", "ec2:DescribeTags", "ec2:DescribeInstances", "ec2:DescribeSubnets", "ec2:DescribeVpcs", "elasticmapreduce:ListInstances", "elasticmapreduce:DescribeCluster", "elasticmapreduce:ListSteps"`
Gunakan kredensyal Git yang disimpan AWS Secrets Manager untuk menautkan repositori Git ke buku catatan.	`"secretsmanager:GetSecretValue"`
Terapkan AWS tag ke antarmuka jaringan dan grup keamanan default	`"ec2:CreateTags"`

Tindakan	Izin
yang dibuat EMR Notebooks saat menyiapkan saluran jaringan aman. Untuk informasi lebih lanjut, lihat Menandai sumber daya AWS.	

Mengakses atau mengunggah file notebook dan metadata ke Amazon S3.	`"s3:PutObject", "s3:GetObject", "s3:GetEncryptionConfiguration", "s3:ListBucket", "s3:DeleteObject"`
	
	Izin berikut hanya diperlukan jika Anda menggunakan bucket Amazon S3 terenkripsi.
	
	`"kms:Decrypt", "kms:GenerateDataKey", "kms:ReEncryptFrom", "kms:ReEncryptTo", "kms:DescribeKey"`

EMR Notebooks memperbarui kebijakan terkelola AWS

Lihat detail tentang pembaruan kebijakan AWS terkelola untuk EMR Notebooks sejak 1 Maret 2021.

Perubahan	Deskripsi	Tanggal
`AmazonElasticMapReduceEditorsRole - Added permissions`	EMR Notebooks `ec2:describeVPCs` ditambahkan `elastmicmapreduce:ListSteps` dan izin untuk. `AmazonElasticMapReduceEditorsRole`	Februari 8, 2023
EMR Notebooks mulai melacak perubahan	EMR Notebooks mulai melacak perubahan untuk AWS kebijakan terkelolanya.	Februari 8, 2023

Menggunakan peran yang berkaitan dengan layanan untuk Amazon EMR

Amazon EMR menggunakan AWS Identity and Access Management (IAM) peran tertaut layanan. Peran tertaut layanan adalah tipe IAM role unik yang ditautkan langsung ke Amazon EMR. Peran tertaut layanan yang telah ditetapkan oleh Amazon EMR dan termasuk izin yang dibutuhkan Amazon EMR untuk memanggil Amazon EC2 atas nama Anda untuk membersihkan sumber daya klaster setelah mereka tidak lagi digunakan. Peran tertaut layanan bekerja sama dengan peran layanan Amazon EMR dan profil instans Amazon EC2 untuk Amazon EMR. Untuk informasi selengkapnya tentang peran layanan dan profil instans, lihat Konfigurasi peran layanan IAM untuk izin Amazon EMR untuk layanan AWS dan sumber daya.

Amazon EMR menentukan izin peran tertaut layanan ini, dan kecuali ditentukan lain, hanya Amazon EMR yang dapat mengambil peran tersebut. Izin yang ditentukan mencakup kebijakan kepercayaan dan kebijakan izin, dan bahwa kebijakan izin tidak dapat dilampirkan ke entitas IAM lainnya. Anda dapat menghapus peran hanya setelah Anda mengakhiri semua klaster EMR di akun.

Untuk informasi tentang layanan lain yang support peran tertaut layanan, lihat AWS layanan yang bekerja dengan IAM dan mencari layanan yang memiliki Ya di kolom Peran Tertaut Layanan. Memilih Ya dengan tautan untuk melihat dokumentasi peran tertaut layanan untuk layanan tersebut.

Izin peran tertaut layanan untuk Amazon EMR

Amazon EMR menggunakan AWSServiceRoleForEMRCleanupperan tersebut, yang merupakan peran berbasis layanan yang memungkinkan Amazon EMR untuk menghentikan dan menghapus sumber daya Amazon EC2 atas nama Anda jika peran layanan Amazon EMR telah kehilangan kemampuan itu. Amazon EMR menciptakan peran secara otomatis selama pembuatan klaster jika tidak sudah ada.

Peran AWSServiceRoleForEMRCleanup terkait layanan mempercayai layanan berikut untuk mengambil peran:

- `elasticmapreduce.amazonaws.com`

Kebijakan izin peran AWSServiceRoleForEMRCleanup tertaut layanan memungkinkan Amazon EMR menyelesaikan tindakan berikut pada sumber daya yang ditentukan:

- Tindakan: `DescribeInstances` pada `ec2`
- Tindakan: `DescribeSpotInstanceRequests` pada `ec2`
- Tindakan: `ModifyInstanceAttribute` pada `ec2`

- Tindakan: `TerminateInstances` pada `ec2`
- Tindakan: `CancelSpotInstanceRequests` pada `ec2`
- Tindakan: `DeleteNetworkInterface` pada `ec2`
- Tindakan: `DescribeInstanceAttribute` pada `ec2`
- Tindakan: `DescribeVolumeStatus` pada `ec2`
- Tindakan: `DescribeVolumes` pada `ec2`
- Tindakan: `DetachVolume` pada `ec2`
- Tindakan: `DeleteVolume` pada `ec2`

Anda harus mengonfigurasikan izin untuk mengizinkan entitas IAM (seperti pengguna, grup, atau peran) untuk membuat, menyunting, atau menghapus peran tertaut layanan.

Untuk mengizinkan entitas IAM membuat peran terkait AWSServiceRoleForEMRCleanup layanan

Menambahkan pernyataan berikut ke kebijakan izin untuk entitas IAM yang perlu membuat peran tertaut layanan.

```
{
    "Effect": "Allow",
    "Action": [
        "iam:CreateServiceLinkedRole",
        "iam:PutRolePolicy"
    ],
    "Resource": "arn:aws:iam::*:role/aws-service-role/elasticmapreduce.amazonaws.com*/
AWSServiceRoleForEMRCleanup*",
    "Condition": {
        "StringLike": {
            "iam:AWSServiceName": [
                "elasticmapreduce.amazonaws.com",
                "elasticmapreduce.amazonaws.com.cn"
            ]
        }
    }
}
```

Untuk mengizinkan entitas IAM mengedit deskripsi peran terkait AWSServiceRoleForEMRCleanup layanan

Menambahkan pernyataan berikut ke kebijakan izin untuk entitas IAM yang perlu mengedit Deskripsi peran tertaut layanan.

```
{
    "Effect": "Allow",
    "Action": [
        "iam:UpdateRoleDescription"
    ],
    "Resource": "arn:aws:iam::*:role/aws-service-role/elasticmapreduce.amazonaws.com*/
AWSServiceRoleForEMRCleanup*",
    "Condition": {
        "StringLike": {
            "iam:AWSServiceName": [
                "elasticmapreduce.amazonaws.com",
                "elasticmapreduce.amazonaws.com.cn"
            ]
        }
    }
}
```

Untuk mengizinkan entitas IAM menghapus peran terkait AWSServiceRoleForEMRCleanup layanan

Menambahkan pernyataan berikut ke kebijakan izin untuk entitas IAM yang perlu menghapus peran tertaut layanan:

```
{
    "Effect": "Allow",
    "Action": [
        "iam:DeleteServiceLinkedRole",
        "iam:GetServiceLinkedRoleDeletionStatus"
    ],
    "Resource": "arn:aws:iam::*:role/aws-service-role/elasticmapreduce.amazonaws.com*/
AWSServiceRoleForEMRCleanup*",
    "Condition": {
        "StringLike": {
            "iam:AWSServiceName": [
                "elasticmapreduce.amazonaws.com",
                "elasticmapreduce.amazonaws.com.cn"
            ]
        }
    }
}
```

Membuat peran tertaut layanan untuk Amazon EMR

Anda tidak perlu membuat AWSServiceRoleForEMRCleanup peran secara manual. Ketika Anda meluncurkan sebuah klaster, baik untuk pertama kalinya atau saat peran tertaut layanan tidak ada, Amazon EMR membuat peran tertaut layanan untuk Anda. Anda harus memiliki izin IAM untuk membuat peran tertaut layanan. Untuk contoh pernyataan yang menambahkan kemampuan ini ke kebijakan izin entitas IAM (seperti pengguna, grup, atau peran), lihat Izin peran tertaut layanan untuk Amazon EMR.

> **Important**
>
> Jika Anda menggunakan Amazon EMR sebelum 24 Oktober 2017, ketika peran terkait layanan tidak didukung, maka Amazon EMR membuat peran tersebut di akun Anda. AWSServiceRoleForEMRCleanup Untuk informasi lebih lanjut, lihat Peran baru yang muncul di akun IAM.

Menyunting peran tertaut layanan untuk Amazon EMR

Amazon EMR tidak memungkinkan Anda mengedit peran terkait AWSServiceRoleForEMRCleanup layanan. Setelah membuat peran tertaut layanan, Anda tidak dapat mengubah nama peran karena berbagai entitas mungkin merujuk peran tersebut. Namun, Anda dapat menyunting Deskripsi peran menggunakan IAM.

Menyunting Deskripsi peran tertaut layanan (konsol IAM)

Anda dapat menggunakan konsol IAM untuk menyunting deskripsi peran tertaut layanan.

Untuk menyunting deskripsi peran tertaut layanan (konsol IAM)

1. Pada panel navigasi konsol IAM, pilih Peran.
2. Memilih nama peran yang akan dimodifikasi.
3. Ke sebelah kanan Deskripsi peranmemilih Sunting.
4. Memasukkan Deskripsi baru di kotak, dan memilih Simpan perubahan.

Mengedit Deskripsi peran tertaut layanan (IAM CLI)

Anda dapat menggunakan perintah IAM dari AWS Command Line Interface untuk mengedit Deskripsi peran tertaut layanan.

Untuk mengubah Deskripsi peran tertaut layanan (CLI)

1. (Opsional) Untuk melihat Deskripsi peran saat ini, gunakan perintah-perintah berikut:

   ```
   $ aws iam get-role --role-name role-name
   ```

   Gunakan nama peran, bukan ARN, untuk merujuk ke peran dengan perintah CLI. Misalnya, jika peran memiliki ARN berikut: `arn:aws:iam::123456789012:role/myrole`, referensi Anda ke peran sebagai **`myrole`**.
2. Untuk memperbarui Deskripsi peran tertaut layanan, gunakan perintah berikut:

   ```
   $ aws iam update-role-description --role-name role-name --description description
   ```

Menyunting Deskripsi peran tertaut layanan (API IAM)

Anda dapat menggunakan IAM API untuk menyunting deskripsi peran tertaut layanan.

Untuk mengubah deskripsi peran tertaut layanan (API)

1. (Opsional) Untuk melihat deskripsi peran saat ini, gunakan perintah berikut:

   API IAM: GetRole
2. Untuk memperbarui deskripsi dari sebuah peran, gunakan perintah berikut:

   API IAM: UpdateRoleDescription

Menghapus peran tertaut layanan untuk Amazon EMR

Jika Anda tidak perlu lagi menggunakan fitur atau layanan yang memerlukan peran tertaut 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 sebelum dapat menghapusnya.

Membersihkan peran tertaut layanan

Sebelum Anda dapat menggunakan IAM untuk menghapus peran tertaut 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. Buka konsol IAM di https://console.aws.amazon.com/iam/.
2. Di panel navigasi, pilih Peran. Pilih nama (bukan kotak centang) AWSServiceRoleForEMRCleanup peran.
3. Di halaman Ringkasan untuk peran yang dipilihmemilih Penasihat Akses.
4. Di tab Penasihat Akses, tinjau aktivitas terbaru untuk peran tertaut layanan.

> **Note**
>
> Jika Anda tidak yakin apakah Amazon EMR menggunakan peran AWSServiceRoleForEMRCleanup tersebut, Anda dapat mencoba menghapus peran tersebut. Jika layanan menggunakan peran tersebut, maka penghapusan akan gagal dan Anda dapat melihat Wilayah tempat peran tersebut digunakan. Jika peran tersebut sedang digunakan, Anda harus menunggu hingga sesi ini berakhir sebelum dapat menghapus peran tersebut. Anda tidak dapat mencabut sesi untuk peran tertaut layanan.

Untuk menghapus sumber daya EMR Amazon yang digunakan oleh AWSServiceRoleForEMRCleanup

- Akhiri semua grup di akun Anda. Untuk informasi selengkapnya, lihat Mengakhiri suatu klaster.

Menghapus peran tertaut layanan (Konsol IAM)

Anda dapat menggunakan konsol IAM untuk menghapus sebuah peran tertaut layanan.

Untuk menghapus peran tertaut layanan (konsol)

1. Buka konsol IAM di https://console.aws.amazon.com/iam/.
2. Di panel navigasi, pilih Peran. Pilih kotak centang di sebelah AWSServiceRoleForEMRCleanup, bukan nama atau baris itu sendiri.
3. Untuk Tindakan peran pada bagian atas halaman, pilih Hapus peran.
4. Pada kotak dialog konfirmasi, tinjau data akses terakhir layanan, yang menunjukkan waktu terakhir setiap peran yang dipilih mengakses layanan AWS. Ini membantu Anda mengonfirmasi aktif tidaknya peran tersebut saat ini. Untuk melanjutkan, pilih Ya, Hapus.

5. Perhatikan notifikasi konsol IAM untuk memantau kemajuan penghapusan peran tertaut layanan. Karena penghapusan peran tertaut layanan IAM bersifat asinkron, setelah Anda kirimkan peran tersebut untuk dihapus, tugas penghapusan dapat berhasil atau gagal. Jika tugas tersebut gagal, Anda dapat memilih Lihat detail atau Lihat Sumber Daya dari notifikasi untuk mempelajari alasan penghapusan gagal. Jika penghapusan gagal karena ada sumber daya di layanan yang digunakan oleh peran tersebut, maka alasan kegagalan tersebut mencakup daftar sumber daya.

Menghapus peran tertaut layanan (IAM CLI)

Anda dapat menggunakan perintah IAM dari AWS Command Line Interface untuk menghapus peran tertaut layanan. Karena peran tertaut layanan tidak dapat dihapus jika sedang digunakan atau memiliki sumber daya terkait, Anda harus kirim permintaan penghapusan. Permintaan tersebut dapat ditolak jika syarat ini tidak terpenuhi.

Untuk menghapus peran tertaut layanan (CLI)

1. Untuk memeriksa status tugas penghapusan, Anda harus menangkap `deletion-task-id` dari tanggapan. Ketik perintah berikut dan kirim permintaan penghapusan peran tertaut layanan:

```
$ aws iam delete-service-linked-role --role-name AWSServiceRoleForEMRCleanup
```

2. Ketik perintah berikut untuk memeriksa status tugas penghapusan:

```
$ aws iam get-service-linked-role-deletion-status --deletion-task-id deletion-task-id
```

   Status tugas penghapusan adalah `NOT_STARTED`, `IN_PROGRESS`, `SUCCEEDED`, atau `FAILED`. Jika penghapusan gagal, panggilan akan mengembalikan alasan kegagalan panggilan sehingga Anda dapat memecahkan masalah.

Menghapus peran tertaut layanan (IAM API)

Anda dapat menggunakan API IAM untuk menghapus peran tertaut layanan. Karena peran tertaut layanan tidak dapat dihapus jika sedang digunakan atau memiliki sumber daya terkait, Anda harus kirim permintaan penghapusan. Permintaan tersebut dapat ditolak jika syarat ini tidak terpenuhi.

Untuk menghapus peran tertaut layanan (API)

1. Untuk mengirimkan permintaan penghapusan peran terkait layanan, hubungi. DeleteServiceLinkedRole Dalam permintaan, tentukan nama AWSServiceRoleForEMRCleanup peran.

   Untuk memeriksa status tugas penghapusan, Anda harus menangkap `DeletionTaskId` dari tanggapan.
2. Untuk memeriksa status penghapusan, panggilGetServiceLinkedRoleDeletionStatus. Pada permintaan itu, tentukan `DeletionTaskId`.

   Status tugas penghapusan adalah `NOT_STARTED`, `IN_PROGRESS`, `SUCCEEDED`, atau `FAILED`. Jika penghapusan gagal, panggilan akan mengembalikan alasan kegagalan panggilan sehingga Anda dapat memecahkan masalah.

Wilayah yang support untuk peran tertaut layanan Amazon EMR

Amazon EMR support penggunaan peran tertaut layanan di Wilayah berikut.

Nama wilayah	Identitas wilayah	Support di Amazon EMR
US East (N. Virginia)	us-east-1	Ya
US East (Ohio)	us-east-2	Ya
US West (N. California)	us-west-1	Ya
US West (Oregon)	us-west-2	Ya
Asia Pacific (Mumbai)	ap-south-1	Ya
Asia Pacific (Osaka)	ap-northeast-3	Ya
Asia Pacific (Seoul)	ap-northeast-2	Ya
Asia Pacific (Singapore)	ap-southeast-1	Ya
Asia Pacific (Sydney)	ap-southeast-2	Ya
Asia Pacific (Tokyo)	ap-northeast-1	Ya

Nama wilayah	Identitas wilayah	Support di Amazon EMR
Canada (Central)	ca-sentral-1	Ya
Eropa (Frankfurt)	eu-central-1	Ya
Eropa (Irlandia)	eu-west-1	Ya
Eropa (London)	eu-west-2	Ya
Europe (Paris)	eu-west-3	Ya
South America (São Paulo)	sa-east-1	Ya

Kustom IAM role

Anda mungkin ingin menyesuaikan peran layanan IAM dan izin untuk membatasi hak sesuai dengan persyaratan keamanan Anda. Untuk menyesuaikan izin, kami merekomendasikan Anda membuat peran dan kebijakan baru. Mulai dengan izin di kebijakan terkelola untuk peran default (misalnya, `AmazonElasticMapReduceforEC2Role` dan `AmazonElasticMapReduceRole`). Kemudian, salin dan tempel konten untuk pernyataan kebijakan baru, modifikasi izin yang sesuai, dan melampirkan kebijakan izin yang sudah diubah untuk peran yang Anda buat. Anda harus memiliki izin IAM yang sesuai untuk bekerja dengan peran dan kebijakan. Untuk informasi selengkapnya, lihat Mengizinkan pengguna dan grup untuk membuat dan memodifikasi peran.

Jika Anda membuat peran EMR kustom untuk EC2, ikuti alur kerja basic, yang secara otomatis membuat profil instans dengan nama yang sama. Amazon EC2 mengizinkan Anda untuk membuat profil instans dan peran dengan nama yang berbeda, tetapi Amazon EMR tidak support konfigurasi ini, dan itu menghasilkan kesalahan "profil instans tidak valid" ketika Anda membuat klaster.

> **Important**
>
> Kebijakan inline tidak diperbarui secara otomatis ketika persyaratan layanan berubah. Jika Anda membuat dan melampirkan kebijakan inline, perhatikan bahwa pembaruan layanan mungkin terjadi yang tiba-tiba menyebabkan kesalahan izin. Untuk informasi lebih lanjut tentang, Kebijakan Terkelola dan Kebijakan Inline di Panduan Pengguna IAM dan Menentukan IAM role kustom ketika Anda membuat sebuah klaster.

Untuk informasi selengkapnya tentang IAM role, lihat topik berikut di bagian Panduan Pengguna IAM:

- Membuat peran untuk mendelegasikan izin ke layanan AWS
- Memodifikasi peran
- Menghapus peran

Menentukan IAM role kustom ketika Anda membuat sebuah klaster

Anda menentukan peran layanan untuk Amazon EMR dan peran untuk profil instans Amazon EC2 ketika Anda membuat sebuah klaster. Pengguna yang menciptakan klaster membutuhkan izin untuk mengambil dan menetapkan peran Amazon EMR dan instans EC2. Jika tidak, akun tidak diizinkan untuk memanggil kesalahan EC2 terjadi. Untuk informasi selengkapnya, lihat Mengizinkan pengguna dan grup untuk membuat dan memodifikasi peran.

Gunakan konsol untuk menentukan peran kustom

Ketika Anda membuat sebuah klaster, Anda dapat menentukan peran layanan kustom untuk Amazon EMR, peran kustom untuk profil instans EC2, dan peran Auto Scaling kustom menggunakan Opsi lanjutan. Saat Anda menggunakan Opsi cepat, peran layanan default dan peran default untuk profil instans EC2 ditentukan. Untuk informasi selengkapnya, lihat Peran layanan IAM yang digunakan oleh Amazon EMR.

> **Note**
>
> Kami telah mendesain ulang konsol EMR Amazon agar lebih mudah digunakan. Lihat Apa yang baru dengan konsol? untuk mempelajari tentang perbedaan antara pengalaman konsol lama dan baru.

New console

Untuk menentukan peran IAM kustom dengan konsol baru

Saat membuat klaster dengan konsol baru, Anda harus menentukan peran layanan khusus untuk Amazon EMR dan peran khusus untuk profil instans EC2. Untuk informasi selengkapnya, lihat Peran layanan IAM yang digunakan oleh Amazon EMR.

1. Masuk keAWS Management Console, dan buka konsol EMR Amazon di https://console.aws.amazon.com/emr.

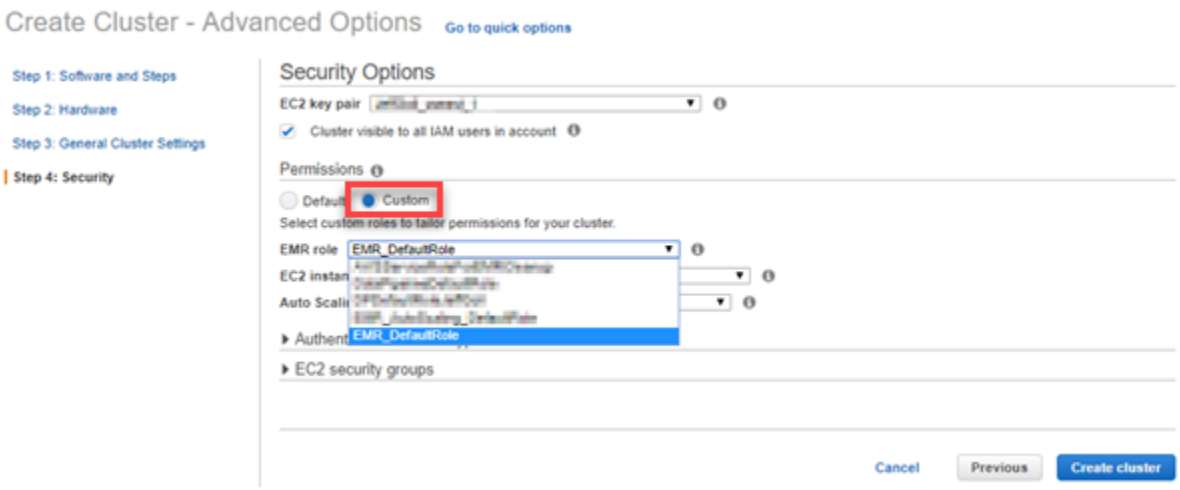
2. Di bawah EMR pada EC2 di panel navigasi kiri, pilih Clusters, lalu pilih Create cluster.
3. Di bawah Konfigurasi dan izin keamanan, temukan peran IAM untuk profil instans dan peran Layanan untuk bidang EMR Amazon. Untuk setiap tipe peran, Anda memilih peran dari daftar. Hanya peran di akun Anda yang memiliki kebijakan kepercayaan yang sesuai untuk tipe peran yang tercantum.
4. Pilih opsi lain yang berlaku untuk cluster Anda.
5. Untuk meluncurkan klaster Anda, pilih Buat klaster.

Old console

Untuk menentukan peran IAM kustom dengan konsol lama

Saat membuat klaster dengan konsol lama, Anda dapat menentukan peran layanan khusus untuk Amazon EMR, peran khusus untuk profil instans EC2, dan peran Auto Scaling khusus menggunakan opsi Lanjutan. Saat Anda menggunakan Opsi cepat, peran layanan default dan peran default untuk profil instans EC2 ditentukan. Untuk informasi selengkapnya, lihat Peran layanan IAM yang digunakan oleh Amazon EMR.

1. Arahkan ke konsol EMR Amazon baru dan pilih Beralih ke konsol lama dari navigasi samping. Untuk informasi selengkapnya tentang apa yang diharapkan saat beralih ke konsol lama, lihat Menggunakan konsol lama.
2. Memilih Buat klaster, Pergi ke opsi lanjutan.
3. Memilih pengaturan klaster yang sesuai untuk aplikasi Anda sampai Anda mencapai Opsi Keamanan. Di bawah Izin, peran Default untuk Amazon EMR dipilih.
4. Memilih Kustom.
5. Untuk setiap tipe peran, Anda memilih peran dari daftar. Hanya peran di akun Anda yang memiliki kebijakan kepercayaan yang sesuai untuk tipe peran yang tercantum.



6. Memilih opsi lain yang sesuai untuk klaster Anda dan lalu memilih Buat klaster.

Gunakan AWS CLI untuk menentukan peran kustom

Anda dapat menentukan peran layanan untuk Amazon EMR dan peran layanan untuk instance EC2 cluster secara eksplisit menggunakan opsi dengan perintah dari. `create-cluster` AWS CLI Gunakan opsi `--service-role` untuk menentukan peran layanan. Gunakan argumen `InstanceProfile` dari opsi `--ec2-attributes` untuk menentukan peran untuk profil instans EC2.

Peran Auto Scaling ditentukan menggunakan opsi terpisah, `--auto-scaling-role`. Untuk informasi selengkapnya, lihat Menggunakan penskalaan otomatis dengan kebijakan kustom untuk grup instans .

Untuk menentukan IAM role kustom menggunakan AWS CLI

- Perintah berikut menentukan peran layanan kustom *`MyCustomServiceRoleForEMR`*, dan peran kustom untuk profil instans EC2 *`MyCustomServiceRoleForClusterEC2Instances`*, saat meluncurkan cluster. Contoh ini menggunakan peran Amazon EMR default.

> **Note**
>
> Karakter lanjutan baris Linux (\) disertakan untuk dibaca. Karakter ini bisa dihapus atau digunakan dalam perintah Linux. Untuk Windows, hapus atau ganti dengan caret (^).

```
aws emr create-cluster --name "Test cluster" --release-label emr-7.0.0 \
```

```
--applications Name=Hive Name=Pig --service-role MyCustomServiceRoleForEMR \
--ec2-attributes InstanceProfile=MyCustomServiceRoleForClusterEC2Instances,\
KeyName=myKey --instance-type m5.xlarge --instance-count 3
```

Anda dapat menggunakan opsi ini untuk menentukan peran default secara eksplisit alih-alih menggunakan opsi `--use-default-roles`. Opsi `--use-default-roles` menentukan peran layanan dan peran untuk profil instans EC2 didefinisikan di file `config` untuk AWS CLI.

Contoh berikut menunjukkan isi dari file `config` untuk AWS CLI yang menentukan peran kustom untuk Amazon EMR. Dengan file konfigurasi ini, ketika `--use-default-roles` opsi ditentukan, cluster dibuat menggunakan *`MyCustomServiceRoleForEMR`*dan *`MyCustomServiceRoleForClusterEC2Instances`*. Secara default, file `config` menentukan default `service_role` sebagai `AmazonElasticMapReduceRole` dan `instance_profile` default sebagai `EMR_EC2_DefaultRole`.

```
[default]
output = json
region = us-west-1
aws_access_key_id = myAccessKeyID
aws_secret_access_key = mySecretAccessKey
emr =
     service_role = MyCustomServiceRoleForEMR
     instance_profile = MyCustomServiceRoleForClusterEC2Instances
```

Konfigurasi IAM role untuk permintaan EMRFS ke Amazon S3

> **Note**
>
> Kemampuan pemetaan peran EMRFS yang dijelaskan di halaman ini telah ditingkatkan dengan diperkenalkannya Hibah Akses Amazon S3 di Amazon EMR 6.15.0. Untuk solusi kontrol akses yang dapat diskalakan untuk data Anda di Amazon S3, sebaiknya gunakan Hibah Akses S3 dengan Amazon EMR.

Ketika aplikasi yang berjalan pada data referensi cluster menggunakan `s3://`*`mydata`* format, Amazon EMR menggunakan EMRFS untuk membuat permintaan. Untuk berinteraksi dengan Amazon S3, EMRFS mengasumsikan kebijakan izin yang dilampirkan ke profil instans Amazon EC2

Anda. Profil instans Amazon EC2 yang sama digunakan terlepas dari pengguna atau grup yang menjalankan aplikasi atau lokasi data di Amazon S3.

Jika Anda memiliki cluster dengan beberapa pengguna yang memerlukan tingkat akses data yang berbeda di Amazon S3 melalui EMRFS, Anda dapat mengatur konfigurasi keamanan dengan peran IAM untuk EMRFS. EMRFS dapat mengambil peran layanan yang berbeda untuk instans EC2 klaster berdasarkan pengguna atau grup yang membuat permintaan, atau berdasarkan lokasi data di Amazon S3. Setiap IAM role untuk EMRFS dapat memiliki izin yang berbeda untuk akses data di Amazon S3. Untuk informasi selengkapnya tentang peran layanan untuk instans EC2 cluster, lihat. Peran layanan untuk instans EC2 klaster (profil instans EC2)

Menggunakan peran IAM khusus untuk EMRFS didukung di Amazon EMR versi 5.10.0 dan yang lebih baru. Jika Anda menggunakan versi yang lebih lama atau memiliki persyaratan di luar peran IAM untuk EMRFS yang disediakan, Anda dapat membuat penyedia kredensyal kustom sebagai gantinya. Untuk informasi selengkapnya, lihat Mengotorisasi akses ke data EMRFS di Amazon S3.

Ketika Anda menggunakan konfigurasi keamanan untuk menentukan IAM role untuk EMRFS, Anda mengatur pemetaan peran. Setiap pemetaan peran menentukan IAM role yang sesuai dengan pengidentifikasi. Pengidentifikasi ini menentukan dasar untuk akses ke Amazon S3 melalui EMRFS. Pengidentifikasi dapat berupa pengguna, grup, atau prefiks Amazon S3 yang menunjukkan lokasi data. Ketika EMRFS membuat permintaan untuk Amazon S3, jika permintaan cocok dengan dasar untuk akses, EMRFS memiliki instans EC2 klaster menganggap IAM role sesuai untuk permintaan. Izin IAM yang terlampir pada peran yang berlaku bukan izin IAM yang terlampir pada peran layanan untuk instans EC2 klaster.

Para pengguna dan grup di pemetaan peran adalah pengguna Hadoop dan grup yang didefinisikan pada klaster. Pengguna dan grup dilewatkan ke EMRFS di konteks aplikasi yang menggunakannya (misalnya, peniruan pengguna YARN). Prefiks Amazon S3 bisa menjadi penspesifikasi bucket dari kedalaman apapun (misalnya, `s3://mybucket` atau `s3://mybucket/myproject/mydata`). Anda dapat menentukan beberapa pengidentifikasi di pemetaan peran tunggal, tetapi mereka semua harus dari tipe yang sama.

> **Important**
>
> IAM role untuk EMRFS menyediakan isolasi tingkat aplikasi antara pengguna aplikasi. Ini tidak menyediakan isolasi tingkat host antara pengguna pada host. Setiap pengguna dengan akses ke klaster dapat melewati isolasi untuk mengambil salah satu peran.

Ketika aplikasi klaster membuat permintaan untuk Amazon S3 melalui EMRFS, EMRFS mengevaluasi pemetaan peran di urutan top-down yang mereka muncul di konfigurasi keamanan. Jika permintaan yang dibuat melalui EMRFS tidak cocok dengan pengidentifikasi apapun, EMRFS akan kembali menggunakan peran layanan untuk instans EC2 klaster. Untuk alasan ini, kami merekomendasikan bahwa kebijakan yang terlampir pada peran ini membatasi izin untuk Amazon S3. Untuk informasi selengkapnya, lihat Peran layanan untuk instans EC2 klaster (profil instans EC2).

Konfigurasi peran

Sebelum Anda mengatur konfigurasi keamanan dengan IAM role untuk EMRFS, rencanakan dan buat kebijakan peran dan izin untuk dilampirkan ke peran. Untuk informasi selengkapnya, lihat Cara kerja peran untuk instans EC2? di Panduan Pengguna IAM. Saat membuat kebijakan izin, sebaiknya Anda memulai dengan kebijakan terkelola yang dilampirkan ke peran EMR Amazon default untuk EC2, lalu mengedit kebijakan ini sesuai dengan kebutuhan Anda. Nama peran default adalah `EMR_EC2_DefaultRole`, dan kebijakan terkelola default untuk mengedit adalah `AmazonElasticMapReduceforEC2Role`. Untuk informasi selengkapnya, lihat Peran layanan untuk instans EC2 klaster (profil instans EC2).

Memperbarui kebijakan kepercayaan untuk mengambil izin peran

Setiap peran yang digunakan EMRFS harus memiliki kebijakan kepercayaan yang memungkinkan peran Amazon EMR klaster untuk EC2 untuk mengasumsikan itu. Demikian pula, peran Amazon EMR cluster untuk EC2 harus memiliki kebijakan kepercayaan yang memungkinkan peran EMRFS untuk mengambilnya.

Kebijakan contoh kepercayaan berikut dilampirkan ke peran untuk EMRFS. Pernyataan tersebut memungkinkan peran EMR Amazon default untuk EC2 untuk mengambil peran tersebut. Misalnya, jika Anda memiliki dua peran EMRFS fiktif, `EMRFSRole_First` dan `EMRFSRole_Second`, pernyataan kebijakan ini ditambahkan ke setiap kebijakan kepercayaan.

```
{
   "Version":"2012-10-17",
   "Statement":[
      {
         "Effect":"Allow",
         "Principal":{
            "AWS":"arn:aws:iam::AWSAcctID:role/EMR_EC2_DefaultRole"
         },
         "Action":"sts:AssumeRole"
      }
```

```
   ]
}
```

Selain itu, contoh pernyataan kebijakan kepercayaan ditambahkan ke `EMR_EC2_DefaultRole` untuk mengizinkan dua peran EMRFS fiktif untuk mengambilnya.

```
{
   "Version":"2012-10-17",
   "Statement":[
      {
         "Effect":"Allow",
         "Principal":{
            "AWS": ["arn:aws:iam::AWSAcctID:role/EMRFSRole_First",
 "arn:aws:iam::AWSAcctID:role/EMRFSRole_Second"]
         },
         "Action":"sts:AssumeRole"
      }
   ]
}
```

Untuk memperbarui kebijakan kepercayaan IAM role

Buka konsol IAM di https://console.aws.amazon.com/iam/.

1. Memilih Peran, memasukkan nama peran di Cari, dan lalu pilih Nama peran.
2. Memilih Hubungan kepercayaan, Sunting hubungan kepercayaan.
3. Tambahkan pernyataan kepercayaan sesuai dengan dokumen Kebijakan sesuai dengan pedoman di atas, lalu pilih Perbarui kebijakan kepercayaan.

Menentukan peran sebagai pengguna kunci

Jika peran memungkinkan akses ke lokasi di Amazon S3 yang dienkripsi menggunakanAWS KMS key, pastikan peran tersebut ditentukan sebagai pengguna kunci. Ini memberikan izin peran untuk menggunakan kunci KMS. Untuk informasi selengkapnya, lihat Kebijakan kunci di AWS KMS di Panduan Developer AWS Key Management Service.

Mengatur konfigurasi keamanan dengan IAM role untuk EMRFS

> **Important**
>
> Jika tidak ada peran IAM untuk EMRFS yang Anda tentukan berlaku, EMRFS kembali ke peran EMR Amazon untuk EC2. Pertimbangkan untuk menyesuaikan peran ini untuk membatasi izin ke Amazon S3 yang sesuai untuk aplikasi Anda dan kemudian menentukan peran kustom ini alih-alih `EMR_EC2_DefaultRole` saat Anda membuat klaster. Untuk informasi lebih lanjut, lihat Kustom IAM role dan Menentukan IAM role kustom ketika Anda membuat sebuah klaster.

Untuk menentukan IAM role untuk permintaan EMRFS ke Amazon S3 menggunakan konsol

1. Membuat konfigurasi keamanan yang menentukan pemetaan peran:
   a. Di konsol Amazon EMR memilih Konfigurasi keamanan, Buat.
   b. Ketik Nama untuk konfigurasi keamanan. Anda menggunakan nama ini untuk menentukan konfigurasi keamanan ketika Anda membuat sebuah klaster.
   c. Memilih Gunakan IAM role untuk permintaan EMRFS ke Amazon S3.
   d. Pilih IAM role untuk meminta, dan di bawah Dasar untuk mengakses memilih tipe pengidentifikasi (Pengguna, Grup, atau prefiks S3) dari daftar dan memasukkan pengidentifikasi yang sesuai. Jika Anda menggunakan beberapa pengidentifikasi, pisahkan dengan koma dan jangan ada spasi. Untuk informasi lebih lanjut tentang setiap tipe pengidentifikasi, lihat JSON configuration reference berikut ini.
   e. Memilih Menambah peran untuk mengatur pemetaan peran tambahan seperti yang dijelaskan di langkah sebelumnya.
   f. Mengatur opsi konfigurasi keamanan lain yang sesuai dan memilih Buat. Untuk informasi selengkapnya, lihat Membuat konfigurasi keamanan.
2. Tentukan konfigurasi keamanan yang Anda buat di atas saat Anda membuat sebuah klaster. Untuk informasi selengkapnya, lihat Menentukan konfigurasi keamanan untuk sebuah klaster.

Untuk menentukan IAM role untuk permintaan EMRFS ke Amazon S3 menggunakan AWS CLI

1. Penggunaan perintah `aws emr create-security-configuration`, menentukan nama untuk konfigurasi keamanan, dan detail konfigurasi keamanan dalam format JSON.

Contoh perintah yang ditunjukkan di bawah ini menciptakan konfigurasi keamanan dengan nama `EMRFS_Roles_Security_Configuration`. Hal ini didasarkan pada struktur JSON di file `MyEmrfsSecConfig.json`, yang disimpan di direktori yang sama dimana perintah dijalankan.

```
aws emr create-security-configuration --name EMRFS_Roles_Security_Configuration --
security-configuration file://MyEmrFsSecConfig.json.
```

Gunakan pedoman berikut untuk struktur file `MyEmrFsSecConfig.json`. Anda dapat menentukan struktur ini bersama dengan struktur untuk opsi konfigurasi keamanan lainnya. Untuk informasi selengkapnya, lihat Membuat konfigurasi keamanan.

Berikut ini adalah contoh potongan JSON untuk menentukan IAM role kustom untuk EMRFS di konfigurasi keamanan. Ini menunjukkan pemetaan peran untuk tiga tipe pengidentifikasi yang berbeda, diikuti dengan referensi parameter.

```
{
  "AuthorizationConfiguration": {
    "EmrFsConfiguration": {
      "RoleMappings": [{
        "Role": "arn:aws:iam::123456789101:role/allow_EMRFS_access_for_user1",
        "IdentifierType": "User",
        "Identifiers": [ "user1" ]
      },{
        "Role": "arn:aws:iam::123456789101:role/allow_EMRFS_access_to_MyBuckets",
        "IdentifierType": "Prefix",
        "Identifiers": [ "s3://MyBucket/","s3://MyOtherBucket/" ]
      },{
        "Role": "arn:aws:iam::123456789101:role/allow_EMRFS_access_for_AdminGroup",
        "IdentifierType": "Group",
        "Identifiers": [ "AdminGroup" ]
      }]
    }
  }
}
```

Parameter	Deskripsi
`"AuthorizationConfiguration":`	Diperlukan.

Parameter	Deskripsi
`"EmrFsConfiguration":`	Diperlukan. Berisi pemetaan peran.
`"RoleMappings":`	Diperlukan. Berisi satu atau lebih definisi peran pemetaan. Pemetaan peran dievaluasi di urutan top-down yang muncul. Jika pemetaan peran mengevaluasi sebagai BETUL untuk panggilan EMRFS untuk data di Amazon S3, tidak ada pemetaan peran lebih lanjut dievaluasi dan EMRFS menggunakan IAM role yang ditentukan untuk permintaan. Pemetaan peran terdiri dari parameter wajib berikut:
`"Role":`	Menentukan pengidentifikasi ARN dari IAM role dalam format `arn:aws:iam::` *`account-id`* `:role/`*`role-name`*. Ini adalah IAM role yang Amazon EMR asumsikan jika permintaan EMRFS ke Amazon S3 cocok dengan salah satu `Identifiers` yang ditentukan.

Parameter	Deskripsi
`"IdentifierType":`	Dapat menjadi salah satu dari yang berikut: • `"User"` menetapkan bahwa pengident ifikasi adalah satu pengguna Hadoop atau lebih, yang bisa saja pengguna akun Linux atau utama Kerberos. Ketika permintaa n EMRFS berasal dari pengguna atau pengguna yang ditentukan, IAM role diasumsikan. • `"Prefix"` menetapkan bahwa pengident ifikasi adalah lokasi Amazon S3. IAM role diasumsikan untuk panggilan ke lokasi atau lokasi dengan prefiks tertentu. Misalnya, prefiks `s3://mybucket/` mencocokkan `s3://mybucket/ mydir` dan `s3://mybucket/yeta notherdir` . • `"Group"` menetapkan bahwa pengident ifikasi adalah satu Grup Hadoop atau lebih. IAM role diasumsikan jika permintaan berasal dari pengguna di grup atau grup-grup tertentu.
`"Identifiers":`	Menentukan satu pengidentifikasi atau lebih dari tipe pengidentifikasi yang sesuai. Pisahkan beberapa pengidentifikasi dengan koma tanpa spasi.

2. Menggunakan perintah `aws emr create-cluster` untuk membuat sebuah klaster dan menentukan konfigurasi keamanan yang Anda buat di langkah sebelumnya.

   Contoh berikut membuat klaster dengan memasang aplikasi Hadoop inti default. Cluster menggunakan konfigurasi keamanan yang dibuat di atas sebagai `EMRFS_Roles_Security_Configuration` dan juga menggunakan peran EMR Amazon

khusus untuk EC2`EC2_Role_EMR_Restrict_S3`, yang ditentukan menggunakan `InstanceProfile` argumen parameter. `--ec2-attributes`

> **Note**
>
> Karakter lanjutan baris Linux (\) disertakan agar mudah dibaca Karakter ini bisa dihapus atau digunakan dalam perintah Linux. Untuk Windows, hapus atau ganti dengan caret (^).

```
aws emr create-cluster --name MyEmrFsS3RolesCluster \
--release-label emr-7.0.0 --ec2-attributes
 InstanceProfile=EC2_Role_EMR_Restrict_S3,KeyName=MyKey \
--instance-type m5.xlarge --instance-count 3 \
--security-configuration EMRFS_Roles_Security_Configuration
```

Gunakan kebijakan berbasis sumber daya untuk akses Amazon EMR ke AWS Katalog Data Glue

Jika Anda menggunakan AWS Glue di hubungannya dengan Hive, Spark, atau Presto di Amazon EMR, AWS Glue mendukung kebijakan berbasis sumber daya untuk mengontrol akses ke sumber daya Katalog Data. Sumber daya ini termasuk database, tabel, koneksi, dan fungsi yang ditetapkan pengguna. Untuk informasi lebih lanjut, lihat AWS Kebijakan sumber daya Glue di AWS Panduan Developer Glue.

Saat menggunakan kebijakan berbasis sumber daya untuk membatasi akses ke AWS Glue dari dalam Amazon EMR, kepala sekolah yang Anda tentukan dalam kebijakan izin harus peran ARN terkait dengan profil contoh EC2 yang ditentukan ketika cluster dibuat. Misalnya, untuk kebijakan berbasis sumber daya yang dilampirkan ke katalog, Anda dapat menentukan peran ARN untuk peran layanan default untuk instance EC2 cluster, *`DefaultRoleEMR_EC2_`* sebagai, menggunakan format yang ditunjukkan dalam contoh berikut: `Principal`

```
arn:aws:iam::acct-id:role/EMR_EC2_DefaultRole
```

Parameter *`ID akt-`* bisa berbeda dari AWS ID akun Glue. Hal ini memungkinkan akses dari cluster EMR di account yang berbeda. Anda dapat menentukan beberapa utama, masing-masing dari akun yang berbeda.

Menggunakan IAM role dengan aplikasi yang memanggil layanan AWS secara langsung

Aplikasi yang berjalan pada instans EC2 klaster dapat menggunakan profil instans EC2 untuk mendapatkan kredensial keamanan sementara ketika memanggil layanan AWS.

Versi Hadoop tersedia dengan Amazon EMR rilis 2.3.0 dan versi terbaru telah diperbarui untuk memanfaatkan IAM role. Jika aplikasi Anda berjalan ketat di atas arsitektur Hadoop, dan tidak langsung memanggil layanan apapun di AWS, maka aplikasi harus bekerja dengan IAM role tanpa modifikasi.

Jika aplikasi Anda memanggil layanan di AWS secara langsung, Anda perlu memperbaruinya untuk memanfaatkan IAM role. Ini berarti bahwa alih-alih mendapatkan kredensial akun dari `/etc/hadoop/conf/core-site.xml` di instans EC2 di klaster, aplikasi Anda menggunakan SDK untuk mengakses sumber daya yang menggunakan IAM role, atau memanggil metadata instans EC2 untuk mendapatkan kredensial sementara.

Untuk mengakses AWS sumber daya dengan IAM role menggunakan SDK

- Topik berikut menunjukkan cara menggunakan beberapa AWS SDK untuk mengakses kredensial sementara menggunakan IAM role. Setiap topik dimulai dengan versi aplikasi yang tidak menggunakan IAM role dan membawa Anda melalui proses mengubah aplikasi untuk menggunakan IAM role.
  - Menggunakan IAM role untuk instans Amazon EC2 dengan SDK for Java di AWS SDK for Java Panduan Developer
  - Menggunakan IAM role untuk instans Amazon EC2 dengan SDK for .NET di AWS SDK for .NET Panduan Developer
  - Menggunakan IAM role untuk instans Amazon EC2 dengan SDK for PHP di AWS SDK for PHP Panduan Developer
  - Menggunakan IAM role untuk instans Amazon EC2 dengan SDK for Ruby di AWS SDK for Ruby Panduan Developer

Untuk mendapatkan kredensial sementara dari metadata instans EC2

- Panggil URL berikut dari instans EC2 yang berjalan dengan peran IAM yang ditentukan, yang mengembalikan kredensyal keamanan sementara terkait (,AccessKeyId, SecretAccessKey

SessionToken, dan Kedaluwarsa). Contoh berikut menggunakan profil instans default untuk Amazon EMR, `EMR_EC2_DefaultRole`.

```
GET http://169.254.169.254/latest/meta-data/iam/security-
credentials/EMR_EC2_DefaultRole
```

Untuk informasi selengkapnya tentang menulis aplikasi yang menggunakan IAM role, lihat Pemberian aplikasi yang berjalan di akses instans Amazon EC2 ke AWS sumber daya.

Untuk informasi lebih lanjut tentang kredensial keamanan sementara, lihat Menggunakan kredensial keamanan sementara di panduan Menggunakan Kredensial Keamanan Sementara.

Mengizinkan pengguna dan grup untuk membuat dan memodifikasi peran

Utama IAM (pengguna dan grup) yang membuat, memodifikasi, dan menentukan peran untuk sebuah klaster, termasuk peran default, harus diizinkan untuk melakukan tindakan berikut. Untuk detail tentang setiap tindakan, lihat Tindakan di Referensi API IAM.

- `iam:CreateRole`
- `iam:PutRolePolicy`
- `iam:CreateInstanceProfile`
- `iam:AddRoleToInstanceProfile`
- `iam:ListRoles`
- `iam:GetPolicy`
- `iam:GetInstanceProfile`
- `iam:GetPolicyVersion`
- `iam:AttachRolePolicy`
- `iam:PassRole`

Izin `iam:PassRole` mengizinkan pembuatan klaster. Izin yang tersisa mengizinkan pembuatan peran default.

Untuk informasi tentang menetapkan izin ke pengguna, lihat Mengubah izin untuk pengguna di Panduan Pengguna IAM.

Kebijakan contoh berbasis identitas Amazon EMR.

Secara default, pengguna dan peran tidak memiliki izin untuk membuat atau memodifikasi sumber daya EMR Amazon. Mereka juga tidak dapat melakukan tugas menggunakan API AWS Management Console, AWS CLI, or AWS. Administrator IAM harus membuat kebijakan IAM yang memberikan izin kepada pengguna dan peran untuk melakukan operasi API tertentu pada sumber daya yang diperlukan. Administrator kemudian harus melampirkan kebijakan tersebut ke pengguna atau grup yang memerlukan izin tersebut.

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

Topik

Praktik terbaik kebijakan untuk Amazon EMR

Kebijakan berbasis identitas sangat kuat. Kebijakan-kebijakan ini menentukan apakah seseorang dapat membuat, mengakses, atau menghapus sumber daya Amazon EMR di akun Anda. Tindakan ini dapat menimbulkan biaya untuk akun AWS Anda. Ketika Anda membuat atau mengedit kebijakan berbasis identitas, ikuti panduan dan rekomendasi ini:

- Mulai Menggunakan AWS Kebijakan Terkelola – Untuk mulai menggunakan Amazon EMR dengan cepat, gunakan kebijakan terkelola AWS untuk memberi karyawan Anda izin yang mereka butuhkan. Kebijakan ini sudah tersedia di akun Anda dan dikelola serta diperbarui oleh AWS. Untuk informasi lebih lanjut, lihat Mulai menggunakan izin dengan AWS kebijakan terkelola di Panduan Pengguna IAM dan Kebijakan terkelola Amazon EMR..
- Berikan Hak Istimewa Minimum – Saat Anda membuat kebijakan khusus, berikan hanya izin yang diperlukan untuk melaksanakan tugas. Mulai dengan satu set izin minimum dan berikan izin tambahan sesuai kebutuhan. Melakukan hal tersebut lebih aman daripada memulai dengan izin

yang terlalu fleksibel, lalu mencoba memperketatnya nanti. Untuk informasi selengkapnya, lihat Pemberian hak istimewa terendah dalam Panduan Pengguna IAM.

- Aktifkan MFA untuk Operasi Sensitif — Untuk keamanan ekstra, pengguna harus menggunakan otentikasi multi-faktor (MFA) untuk mengakses sumber daya sensitif atau operasi API. Untuk informasi selengkapnya, lihat Menggunakan autentikasi multifaktor (MFA) dalam AWS dalam Panduan Pengguna IAM.
- Gunakan Kondisi Kebijakan untuk Keamanan Tambahan – Selama praktis, tentukan ketentuan di mana kebijakan berbasis identitas Anda memungkinkan akses ke sumber daya. Misalnya, Anda dapat menulis persyaratan untuk menentukan jangkauan alamat IP yang diizinkan untuk mengajukan permintaan. Anda juga dapat menulis persyaratan untuk mengizinkan permintaan hanya dalam rentang tanggal atau waktu tertentu, atau untuk mewajibkan penggunaan SSL atau autentikasi multifaktor (MFA). Untuk informasi lebih lanjut, lihat Elemen kebijakan IAM JSON: Syarat di Panduan Pengguna IAM.

Izinkan pengguna untuk melihat izin mereka sendiri

Contoh ini menunjukkan cara Anda membuat kebijakan yang memungkinkan pengguna melihat kebijakan sebaris dan terkelola yang dilampirkan pada identitas pengguna mereka. Kebijakan ini mencakup izin untuk menyelesaikan tindakan ini pada konsol atau secara terprogram menggunakan AWS CLI atau API AWS.

```
{
   "Version":"2012-10-17",
   "Statement":[
      {
         "Sid":"ViewOwnUserInfo",
         "Effect":"Allow",
         "Action":[
            "iam:GetUser",
            "iam:GetUserPolicy",
            "iam:ListAttachedUserPolicies",
            "iam:ListGroupsForUser",
            "iam:ListUserPolicies"
         ],
         "Resource":[
            "arn:aws:iam:::user/${aws:username}"
         ]
      },
      {
```

```
         "Sid":"NavigateInConsole",
         "Effect":"Allow",
         "Action":[
            "iam:GetGroupPolicy",
            "iam:GetPolicy",
            "iam:GetPolicyVersion",
            "iam:ListAttachedGroupPolicies",
            "iam:ListGroupPolicies",
            "iam:ListGroups",
            "iam:ListPolicies",
            "iam:ListPolicyVersions",
            "iam:ListUsers"
         ],
         "Resource":""
      }
   ]
}
```

Kebijakan terkelola Amazon EMR.

Cara termudah untuk memberikan akses penuh atau akses hanya-baca untuk tindakan Amazon EMR yang diperlukan adalah dengan menggunakan kebijakan terkelola IAM untuk Amazon EMR. Kebijakan terkelola menawarkan manfaat pemutakhiran secara otomatis jika persyaratan izin berubah. Jika Anda menggunakan kebijakan inline, perubahan layanan dapat terjadi yang menyebabkan kesalahan izin muncul.

Amazon EMR akan menghentikan kebijakan terkelola yang ada (kebijakan v1) demi kebijakan terkelola baru (kebijakan v2). Kebijakan terkelola baru telah dicakup untuk menyelaraskan dengan praktik terbaik AWS. Setelah kebijakan terkelola v1 yang ada tidak digunakan lagi, Anda tidak akan dapat melampirkan kebijakan ini ke peran atau pengguna IAM baru. Peran dan pengguna yang ada yang menggunakan kebijakan yang tidak lagi digunakan dapat terus menggunakannya. Kebijakan terkelola v2 membatasi akses menggunakan tag. Mereka hanya mengizinkan tindakan EMR Amazon yang ditentukan dan memerlukan sumber daya cluster yang ditandai dengan kunci khusus EMR. Kami merekomendasikan bahwa Anda hati-hati meninjau dokumentasi sebelum menggunakan kebijakan v2 baru.

Kebijakan v1 tidak lagi digunakan dengan ikon pemberitahuan di sebelahnya di daftar Kebijakan di konsol IAM. Kebijakan yang tidak lagi digunakan akan memiliki karakteristik sebagai berikut:

- Kebijakan terus berfungsi untuk semua pengguna, grup, dan peran yang dilampirkan saat ini. Tidak ada yang rusak.

- Kebijakan tidak dapat dilampirkan pada pengguna, grup, atau peran baru. Jika Anda melepas salah satu kebijakan dari entitas yang ada, Anda tidak dapat melampirkannya kembali.
- Setelah Anda melepaskan kebijakan v1 dari semua entitas saat ini, kebijakan tidak lagi akan terlihat dan tidak lagi dapat digunakan.

Tabel berikut merangkum perubahan antara kebijakan saat ini (v1) dan kebijakan v2.

Perubahan kebijakan terkelola EMR

Tipe kebijakan	Nama kebijakan	Tujuan kebijakan	Perubahan kebijakan v2
Kebijakan terkelola IAM untuk akses EMR penuh oleh pengguna, peran, atau grup terlampir	Kebijakan V1 (tidak digunakan lagi): **AmazonElasticMapReduceFullAccess** Nama kebijakan V2 (cakupan): **AmazonEMRFullAccessPolicy_v2**	Mengizinkan izin penuh pengguna untuk tindakan EMR. Termasuk iam: PassRole izin untuk sumber daya.	Kebijakan menambahkan prasyarat bahwa pengguna harus menambahkan tanda pengguna ke sumber daya sebelum mereka dapat menggunakan kebijakan ini. Lihat Penandaan sumber daya untuk menggunakan kebijakan terkelola. iam: PassRole tindakan membutuhkan iam: PassedToService kondisi disetel ke layanan tertentu. Akses ke Amazon EC2, Amazon S3, dan layanan lainnya tidak diizinkan secara default. Lihat Kebijakan Terkelola

Tipe kebijakan	Nama kebijakan	Tujuan kebijakan	Perubahan kebijakan v2
			IAM untuk Akses Penuh (Kebijakan Default Terkelola v2).
Kebijakan terkelola IAM untuk akses hanya-baca oleh pengguna, peran, atau grup terlampir	Kebijakan V1 (tidak digunakan lagi): **AmazonElasticMapReduceReadOnlyAccess** Nama kebijakan V2 (cakupan): **AmazonEMRReadOnlyAccessPolicy_v2**	Mengizinkan izin hanya-baca pengguna untuk tindakan Amazon EMR.	Izin hanya mengizinkan tindakan hanya-baca elasticmapreduce yang ditentukan. Akses ke Amazon S3 adalah akses yang tidak diizinkan secara default. Lihat Kebijakan Terkelola IAM untuk Akses Hanya-Baca (Kebijakan Default Terkelola v2).

Tipe kebijakan	Nama kebijakan	Tujuan kebijakan	Perubahan kebijakan v2
Peran layanan EMR default dan kebijakan terkelola terlampir	Nama peran: EMR_DefaultRole Kebijakan V1 (tidak digunakan lagi): (Peran Layanan AmazonElasticMapReduceRoleEMR) Nama kebijakan (dicakup) V2: **AmazonEMRServicePolicy_v2**	Mengizinkan Amazon EMR untuk memanggil layanan AWS lain atas nama Anda saat menyediakan sumber daya dan melakukan tindakan tingkat layanan. Peran ini diperlukan untuk semua klaster.	Peran layanan v2 dan kebijakan default v2 menggantikan peran dan kebijakan yang tidak lagi digunakan. Kebijakan menambahkan prasyarat bahwa pengguna harus menambahkan tanda pengguna ke sumber daya sebelum mereka dapat menggunakan kebijakan ini. Lihat Penandaan sumber daya untuk menggunakan kebijakan terkelola. Lihat Peran layanan untuk Amazon EMR (peran EMR).

Tipe kebijakan	Nama kebijakan	Tujuan kebijakan	Perubahan kebijakan v2
Peran layanan untuk instans EC2 klaster (profil instans EC2)	Kebijakan V1 (tidak digunakan lagi): DefaultRoleEMR_EC2_ (profil contoh) Nama kebijakan usang: EC2role AmazonElasticMapReducefor	Mengizinkan aplikasi yang berjalan pada klaster EMR untuk mengakses sumber daya AWS lain, seperti Amazon S3. Misalnya, jika Anda menjalankan tugas Apache Spark yang memproses data dari Amazon S3, kebijakan perlu mengizinkan akses ke sumber daya tersebut.	Peran default dan kebijakan default berada di jalur yang tidak lagi digunakan. Tidak ada pengganti AWS peran atau kebijakan terkelola default. Anda harus memberikan kebijakan berbasis sumber daya atau kebijakan berbasis identitas. Ini berarti bahwa, secara default, aplikasi yang berjalan pada klaster EMR tidak memiliki akses ke Amazon S3 atau sumber daya lain kecuali Anda secara manual menambahkan ini ke kebijakan. Lihat Peran default dan kebijakan terkelola.

Tipe kebijakan	Nama kebijakan	Tujuan kebijakan	Perubahan kebijakan v2
Kebijakan peran layanan EC2 lainnya	Nama kebijakan saat ini: AmazonElasticMapReduceforAutoScalingRole,AmazonElasticMapReduceEditorsRole, AmazonEMR CleanupPolicy	Memberikan izin yang dibutuhkan EMR untuk mengakses sumber daya AWS lain dan melakukan tindakan jika menggunakan penskalaan otomatis, notebook, atau untuk membersihkan sumber daya EC2.	Tidak ada perubahan untuk v2.

Mengamankan iam: PassRole

Kebijakan terkelola default izin penuh Amazon EMR menggabungkan konfigurasi `iam:PassRole` keamanan, termasuk yang berikut ini:

- Izin `iam:PassRole` hanya untuk peran Amazon EMR default tertentu.
- `iam:PassedToService`kondisi yang memungkinkan Anda untuk menggunakan kebijakan hanya dengan AWS layanan tertentu, seperti `elasticmapreduce.amazonaws.com` dan`ec2.amazonaws.com`.

Anda dapat melihat versi JSON dari kebijakan AmazonEMR FullAccessPolicy _v2 dan ServicePolicyAmazonEMR _v2 di konsol IAM. Kami menyarankan Anda membuat klaster baru dengan kebijakan terkelola v2.

Untuk membuat kebijakan khusus, kami merekomendasikan sebaiknya Anda mulai dengan kebijakan terkelola dan mengeditnya sesuai dengan kebutuhan Anda.

Untuk informasi tentang cara melampirkan kebijakan ke pengguna (prinsipal), lihat Bekerja dengan kebijakan terkelola menggunakan Panduan Pengguna IAM. AWS Management Console

Penandaan sumber daya untuk menggunakan kebijakan terkelola

AmazonEMR ServicePolicy _v2 dan AmazonEMR _v2 bergantung pada akses tercakup ke FullAccessPolicy sumber daya yang disediakan atau digunakan Amazon EMR. Cakupan ke bawah dicapai dengan membatasi akses hanya ke sumber daya yang memiliki tag pengguna yang telah ditentukan sebelumnya yang terkait dengannya. Bila Anda menggunakan salah satu dari dua kebijakan ini, Anda harus meneruskan tag pengguna yang telah ditentukan `for-use-with-amazon-emr-managed-policies = true` saat Anda menyediakan klaster. Amazon EMR kemudian akan secara otomatis menyebarkan tanda itu. Selain itu, Anda harus menambahkan tag pengguna ke sumber daya yang tercantum di bagian berikut. Jika Anda menggunakan konsol EMR Amazon untuk meluncurkan cluster Anda, lihat. Pertimbangan untuk menggunakan konsol EMR Amazon untuk meluncurkan cluster dengan kebijakan terkelola v2

Untuk menggunakan kebijakan terkelola, teruskan tag pengguna `for-use-with-amazon-emr-managed-policies = true` saat Anda menyediakan klaster dengan CLI, SDK, atau metode lain.

Ketika Anda melewatkan tanda, Amazon EMR menyebarkan tanda untuk ENI subnet privat, instans EC2, dan volume EBS yang dibuatnya. Amazon EMR juga secara otomatis memberi tanda grup keamanan yang dibuat. Namun, jika Anda ingin Amazon EMR untuk diluncurkan dengan grup keamanan tertentu, Anda harus memberinya tanda. Untuk sumber daya yang tidak dibuat oleh Amazon EMR, Anda harus menambahkan tag ke sumber daya tersebut. Misalnya, Anda harus menandai subnet Amazon EC2, grup keamanan EC2 (jika tidak dibuat oleh Amazon EMR), dan VPC (jika Anda ingin Amazon EMR membuat grup keamanan). Untuk meluncurkan cluster dengan kebijakan terkelola v2 di VPC, Anda harus menandai VPC tersebut dengan tag pengguna yang telah ditentukan. Lihat, Pertimbangan untuk menggunakan konsol EMR Amazon untuk meluncurkan cluster dengan kebijakan terkelola v2.

Penyebaran penandaan yang ditentukan pengguna

Sumber daya tanda Amazon EMR yang dibuat menggunakan tanda Amazon EMR yang Anda tentukan saat membuat klaster. Amazon EMR membuat tanda untuk sumber daya yang dibuat selama masa klaster.

Amazon EMR menyebarkan tanda pengguna untuk sumber daya berikut:

- ENI Subnet Privat (antarmuka jaringan elastis akses layanan)
- Instans EC2
- Volume EBS

- Templat Peluncuran EC2

Grup keamanan bertanda otomatis

Amazon EMR menandai grup keamanan EC2 yang menciptakan dengan tanda yang diperlukan untuk kebijakan terkelola v2 untuk Amazon EMR, `for-use-with-amazon-emr-managed-policies`, terlepas dari tanda yang Anda tentukan di perintah buat klaster. Untuk grup keamanan yang dibuat sebelum pengidentifikasian kebijakan terkelola v2, Amazon EMR tidak secara otomatis memberi tanda pada grup keamanan. Jika Anda ingin menggunakan kebijakan terkelola v2 dengan grup keamanan default yang sudah ada di akun, Anda perlu memberi tanda pada grup keamanan secara manual dengan `for-use-with-amazon-emr-managed-policies = true`.

Sumber daya klaster yang beri tanda secara manual

Anda harus secara manual memberi tanda pada beberapa sumber klaster sehingga mereka dapat diakses oleh peran default Amazon EMR.

- Anda harus secara manual memberi tanda pada grup keamanan EC2 dan subnet EC2 dengan tanda kebijakan terkelola Amazon EMR `for-use-with-amazon-emr-managed-policies`.
- Anda harus secara manual memberi tanda pada VPC jika Anda ingin Amazon EMR untuk membuat grup keamanan default. EMR akan mencoba membuat grup keamanan dengan tanda tertentu jika grup keamanan default belum ada.

Amazon EMR secara otomatis memberi tanda pada sumber daya berikut:

- Grup Keamanan EC2 yang dibuat oleh EMR

Anda harus secara manual memberi tanda pada sumber daya berikut:

- Subnet EC2
- Grup Keamanan EC2

Opsional, Anda dapat secara manual memberi tanda pada sumber daya berikut:

- VPC - hanya bila Anda ingin Amazon EMR untuk membuat grup keamanan

Pertimbangan untuk menggunakan konsol EMR Amazon untuk meluncurkan cluster dengan kebijakan terkelola v2

Anda dapat menyediakan kluster dengan kebijakan terkelola v2 menggunakan konsol EMR Amazon. Berikut adalah beberapa pertimbangan saat Anda menggunakan konsol untuk meluncurkan cluster EMR Amazon.

> **Note**
>
> Kami telah mendesain ulang konsol EMR Amazon. Kemampuan penandaan otomatis belum tersedia di konsol baru, dan konsol baru juga tidak menunjukkan sumber daya (VPC/subnet) mana yang perlu diberi tag. Lihat Apa yang baru dengan konsol? untuk mempelajari lebih lanjut tentang perbedaan antara pengalaman konsol lama dan baru.

- Anda tidak perlu melewati tag yang telah ditentukan. Amazon EMR secara otomatis menambahkan tag dan menyebarkannya ke komponen yang sesuai.
- Untuk komponen yang perlu diberi tag secara manual, konsol EMR Amazon lama mencoba memberi tag secara otomatis jika Anda memiliki izin yang diperlukan untuk menandai sumber daya. Jika Anda tidak memiliki izin untuk menandai sumber daya atau jika Anda ingin menggunakan konsol baru, minta administrator untuk menandai sumber daya tersebut.
- Anda tidak dapat meluncurkan cluster dengan kebijakan terkelola v2 kecuali semua prasyarat terpenuhi.
- Konsol EMR Amazon lama menunjukkan sumber daya (VPC/subnet) mana yang perlu diberi tag.

Kebijakan terkelola IAM untuk akses penuh (kebijakan default terkelola v2)

Kebijakan terkelola default EMR yang dicakup v2 memberikan hak istimewa akses khusus kepada pengguna. Mereka membutuhkan tanda sumber daya Amazon EMR yang telah ditetapkan dan kunci syarat `iam:PassRole` untuk sumber daya yang digunakan oleh Amazon EMR, seperti `Subnet` dan `SecurityGroup` yang Anda gunakan untuk meluncurkan klaster Anda.

Untuk memberikan cakupan tindakan yang diperlukan untuk Amazon EMR, melampirkan `AmazonEMRFullAccessPolicy_v2` kebijakan terkelola. Kebijakan terkelola default yang diperbarui ini menggantikan `AmazonElasticMapReduceFullAccess` kebijakan terkelola.

`AmazonEMRFullAccessPolicy_v2` tergantung pada akses yang dicakup ke sumber daya yang disediakan atau digunakan Amazon EMR. Bila menggunakan kebijakan ini, Anda harus

melewati tanda pengguna `for-use-with-amazon-emr-managed-policies = true` saat menyediakan klaster. Amazon EMR secara otomatis akan menyebarkan tanda. Selain itu, Anda mungkin perlu secara manual menambahkan tanda pengguna untuk tipe sumber daya tertentu, seperti grup keamanan EC2 yang tidak dibuat oleh Amazon EMR. Untuk informasi selengkapnya, lihat Penandaan sumber daya untuk menggunakan kebijakan terkelola.

`AmazonEMRFullAccessPolicy_v2`Kebijakan mengamankan sumber daya dengan melakukan hal berikut:

- Memerlukan sumber daya yang akan ditandai dengan tanda kebijakan terkelola Amazon EMR yang telah ditetapkan `for-use-with-amazon-emr-managed-policies` untuk pembuatan klaster dan akses Amazon EMR.
- Membatasi tindakan `iam:PassRole` untuk peran default tertentu dan akses `iam:PassedToService` ke layanan tertentu.
- Tidak lagi menyediakan akses ke Amazon EC2, Amazon S3, dan layanan lainnya secara default.

Berikut ini adalah isi dari kebijakan ini.

> **Note**
>
> Anda juga dapat menggunakan tautan konsol `AmazonEMRFullAccessPolicy_v2` untuk melihat kebijakan.

```
{
    "Version": "2012-10-17",
    "Statement": [
        {
            "Sid": "RunJobFlowExplicitlyWithEMRManagedTag",
            "Effect": "Allow",
            "Action": [
                "elasticmapreduce:RunJobFlow"
            ],
            "Resource": "*",
            "Condition": {
                "StringEquals": {
                    "aws:RequestTag/for-use-with-amazon-emr-managed-policies": "true"
                }
            }
```

```
        },
        {
            "Sid": "ElasticMapReduceActions",
            "Effect": "Allow",
            "Action": [
                "elasticmapreduce:AddInstanceFleet",
                "elasticmapreduce:AddInstanceGroups",
                "elasticmapreduce:AddJobFlowSteps",
                "elasticmapreduce:AddTags",
                "elasticmapreduce:CancelSteps",
                "elasticmapreduce:CreateEditor",
                "elasticmapreduce:CreateSecurityConfiguration",
                "elasticmapreduce:DeleteEditor",
                "elasticmapreduce:DeleteSecurityConfiguration",
                "elasticmapreduce:DescribeCluster",
                "elasticmapreduce:DescribeEditor",
                "elasticmapreduce:DescribeJobFlows",
                "elasticmapreduce:DescribeSecurityConfiguration",
                "elasticmapreduce:DescribeStep",
                "elasticmapreduce:DescribeReleaseLabel",
                "elasticmapreduce:GetBlockPublicAccessConfiguration",
                "elasticmapreduce:GetManagedScalingPolicy",
                "elasticmapreduce:GetAutoTerminationPolicy",
                "elasticmapreduce:ListBootstrapActions",
                "elasticmapreduce:ListClusters",
                "elasticmapreduce:ListEditors",
                "elasticmapreduce:ListInstanceFleets",
                "elasticmapreduce:ListInstanceGroups",
                "elasticmapreduce:ListInstances",
                "elasticmapreduce:ListSecurityConfigurations",
                "elasticmapreduce:ListSteps",
                "elasticmapreduce:ListSupportedInstanceTypes",
                "elasticmapreduce:ModifyCluster",
                "elasticmapreduce:ModifyInstanceFleet",
                "elasticmapreduce:ModifyInstanceGroups",
                "elasticmapreduce:OpenEditorInConsole",
                "elasticmapreduce:PutAutoScalingPolicy",
                "elasticmapreduce:PutBlockPublicAccessConfiguration",
                "elasticmapreduce:PutManagedScalingPolicy",
                "elasticmapreduce:RemoveAutoScalingPolicy",
                "elasticmapreduce:RemoveManagedScalingPolicy",
                "elasticmapreduce:RemoveTags",
                "elasticmapreduce:SetTerminationProtection",
                "elasticmapreduce:StartEditor",
```

```
                "elasticmapreduce:StopEditor",
                "elasticmapreduce:TerminateJobFlows",
                "elasticmapreduce:ViewEventsFromAllClustersInConsole"
            ],
            "Resource": "*"
        },
        {
            "Sid": "ViewMetricsInEMRConsole",
            "Effect": "Allow",
            "Action": [
                "cloudwatch:GetMetricStatistics"
            ],
            "Resource": "*"
        },
        {
            "Sid": "PassRoleForElasticMapReduce",
            "Effect": "Allow",
            "Action": "iam:PassRole",
            "Resource": [
                "arn:aws:iam::*:role/EMR_DefaultRole",
                "arn:aws:iam::*:role/EMR_DefaultRole_V2"
            ],
            "Condition": {
                "StringLike": {
                    "iam:PassedToService": "elasticmapreduce.amazonaws.com*"
                }
            }
        },
        {
            "Sid": "PassRoleForEC2",
            "Effect": "Allow",
            "Action": "iam:PassRole",
            "Resource": "arn:aws:iam::*:role/EMR_EC2_DefaultRole",
            "Condition": {
                "StringLike": {
                    "iam:PassedToService": "ec2.amazonaws.com*"
                }
            }
        },
        {
            "Sid": "PassRoleForAutoScaling",
            "Effect": "Allow",
            "Action": "iam:PassRole",
            "Resource": "arn:aws:iam::*:role/EMR_AutoScaling_DefaultRole",
```

```
            "Condition": {
                "StringLike": {
                    "iam:PassedToService": "application-autoscaling.amazonaws.com*"
                }
            }
        },
        {
            "Sid": "ElasticMapReduceServiceLinkedRole",
            "Effect": "Allow",
            "Action": "iam:CreateServiceLinkedRole",
            "Resource": "arn:aws:iam::*:role/aws-service-role/
elasticmapreduce.amazonaws.com*/AWSServiceRoleForEMRCleanup*",
            "Condition": {
                "StringEquals": {
                    "iam:AWSServiceName": [
                        "elasticmapreduce.amazonaws.com",
                        "elasticmapreduce.amazonaws.com.cn"
                    ]
                }
            }
        },
        {
            "Sid": "ConsoleUIActions",
            "Effect": "Allow",
            "Action": [
                "ec2:DescribeAccountAttributes",
                "ec2:DescribeAvailabilityZones",
                "ec2:DescribeImages",
                "ec2:DescribeKeyPairs",
                "ec2:DescribeNatGateways",
                "ec2:DescribeRouteTables",
                "ec2:DescribeSecurityGroups",
                "ec2:DescribeSubnets",
                "ec2:DescribeVpcs",
                "ec2:DescribeVpcEndpoints",
                "s3:ListAllMyBuckets",
                "iam:ListRoles"
            ],
            "Resource": "*"
        }
    ]
}
```

Kebijakan terkelola IAM untuk akses penuh (pada jalur yang tidak lagi digunakan)

Kebijakan terkelola `AmazonElasticMapReduceFullAccess` dan `AmazonEMRFullAccessPolicy_v2` AWS Identity and Access Management (IAM) memberikan semua tindakan yang diperlukan untuk Amazon EMR dan layanan lainnya.

> **Important**
>
> Kebijakan `AmazonElasticMapReduceFullAccess` terkelola berada di jalur menuju penghentian, dan tidak lagi direkomendasikan untuk digunakan dengan Amazon EMR. Sebaliknya, gunakan `AmazonEMRFullAccessPolicy_v2`. Ketika layanan IAM akhirnya menghentikan kebijakan v1, Anda tidak akan dapat melampirkannya ke peran. Namun, Anda dapat melampirkan peran yang ada ke klaster meskipun peran tersebut menggunakan kebijakan yang tidak digunakan lagi.

Kebijakan terkelola default izin penuh Amazon EMR menggabungkan konfigurasi `iam:PassRole` keamanan, termasuk yang berikut ini:

- Izin `iam:PassRole` hanya untuk peran Amazon EMR default tertentu.
- `iam:PassedToService`kondisi yang memungkinkan Anda untuk menggunakan kebijakan hanya dengan AWS layanan tertentu, seperti `elasticmapreduce.amazonaws.com` dan`ec2.amazonaws.com`.

Anda dapat melihat versi JSON dari kebijakan AmazonEMR FullAccessPolicy _v2 dan ServicePolicyAmazonEMR _v2 di konsol IAM. Kami menyarankan Anda membuat klaster baru dengan kebijakan terkelola v2.

Anda dapat melihat konten kebijakan v1 yang tidak digunakan lagi di at. AWS Management Console `AmazonElasticMapReduceFullAccess` `ec2:TerminateInstances`Tindakan dalam kebijakan memberikan izin kepada pengguna atau peran untuk menghentikan instans Amazon EC2 yang terkait dengan akun IAM. Ini termasuk contoh yang bukan bagian dari cluster EMR.

Kebijakan terkelola IAM untuk akses hanya-baca (kebijakan default terkelola v2)

Untuk memberikan hak istimewa hanya-baca ke Amazon EMR, lampirkan kebijakan terkelola AmazonEMR _v2. ReadOnlyAccessPolicy Kebijakan terkelola default ini menggantikan kebijakan terkelola `AmazonElasticMapReduceReadOnlyAccess`.

Isi dari pernyataan kebijakan ini ditampilkan di potongan berikut. Dibandingkan dengan kebijakan `AmazonElasticMapReduceReadOnlyAccess`, kebijakan `AmazonEMRReadOnlyAccessPolicy_v2` tidak menggunakan karakter wildcard untuk elemen `elasticmapreduce`. Sebagai gantinya, kebijakan v2 default mencakup tindakan yang diizinkan`elasticmapreduce`.

> **Note**
>
> Anda juga dapat menggunakan tautan AWS Management Console [AmazonEMRReadOnlyAccessPolicy_v2] untuk melihat kebijakan.

```
{
    "Version": "2012-10-17",
    "Statement": [
        {
            "Sid": "ElasticMapReduceActions",
            "Effect": "Allow",
            "Action": [
                "elasticmapreduce:DescribeCluster",
                "elasticmapreduce:DescribeEditor",
                "elasticmapreduce:DescribeJobFlows",
                "elasticmapreduce:DescribeSecurityConfiguration",
                "elasticmapreduce:DescribeStep",
                "elasticmapreduce:DescribeReleaseLabel",
                "elasticmapreduce:GetBlockPublicAccessConfiguration",
                "elasticmapreduce:GetManagedScalingPolicy",
                "elasticmapreduce:GetAutoTerminationPolicy",
                "elasticmapreduce:ListBootstrapActions",
                "elasticmapreduce:ListClusters",
                "elasticmapreduce:ListEditors",
                "elasticmapreduce:ListInstanceFleets",
                "elasticmapreduce:ListInstanceGroups",
                "elasticmapreduce:ListInstances",
                "elasticmapreduce:ListSecurityConfigurations",
                "elasticmapreduce:ListSteps",
                "elasticmapreduce:ListSupportedInstanceTypes",
                "elasticmapreduce:ViewEventsFromAllClustersInConsole"
            ],
            "Resource": "*"
        },
        {
```

```
            "Sid": "ViewMetricsInEMRConsole",
            "Effect": "Allow",
            "Action": [
                "cloudwatch:GetMetricStatistics"
            ],
            "Resource": "*"
        }
    ]
}
```

Kebijakan terkelola IAM untuk akses hanya-baca (di jalur yang tidak lagi digunakan)

Kebijakan terkelola `AmazonElasticMapReduceReadOnlyAccess` berada di jalur yang tidak lagi digunakan. Anda tidak dapat melampirkan kebijakan ini ketika meluncurkan klaster baru. `AmazonElasticMapReduceReadOnlyAccess` telah diganti dengan `AmazonEMRReadOnlyAccessPolicy_v2` sebagai kebijakan terkelola default Amazon EMR. Isi dari pernyataan kebijakan ini ditampilkan di potongan berikut. Karakter wildcard untuk elemen `elasticmapreduce` menentukan bahwa hanya tindakan yang dimulai dengan string tertentu diizinkan. Perlu diingat bahwa karena kebijakan ini tidak secara eksplisit menolak tindakan, pernyataan kebijakan yang berbeda masih dapat digunakan untuk memberikan akses ke tindakan tertentu.

> **Note**
>
> Anda juga dapat menggunakan AWS Management Console untuk melihat kebijakan.

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Effect": "Allow",
      "Action": [
        "elasticmapreduce:Describe*",
        "elasticmapreduce:List*",
        "elasticmapreduce:ViewEventsFromAllClustersInConsole",
        "s3:GetObject",
        "s3:ListAllMyBuckets",
        "s3:ListBucket",
        "sdb:Select",
        "cloudwatch:GetMetricStatistics"
```

```
      ],
      "Resource": "*"
    }
  ]
}
```

Kebijakan terkelola AWS untuk Amazon EMR

Untuk menambahkan izin ke pengguna, grup, dan peran, lebih mudah menggunakan kebijakan terkelola AWS daripada menulis kebijakan sendiri. Dibutuhkan waktu dan keahlian untuk membuat kebijakan terkelola pelanggan IAM yang hanya menyediakan izin sesuai kebutuhan tim Anda. Untuk mulai dengan cepat, Anda dapat menggunakan kebijakan-kebijakan terkelola AWS kami. Kebijakan-kebijakan ini mencakup kasus penggunaan umum dan tersedia di akun AWS Anda. Untuk informasi lebih lanjut tentang kebijakan-kebijakan terkelola AWS, lihat kebijakan terkelola AWS di Panduan Pengguna IAM.

Layanan AWS mempertahankan dan memperbarui kebijakan-kebijakan terkelola AWS. Anda tidak dapat mengubah izin yang ada dalam kebijakan-kebijakan yang dikelola AWS. Layanan terkadang menambahkan izin tambahan ke kebijakan yang dikelola AWS untuk mendukung fitur-fitur baru. Jenis pembaruan ini akan memengaruhi semua identitas (pengguna, grup, dan peran) di mana kebijakan tersebut dilampirkan. Layanan kemungkinan besar akan memperbarui kebijakan yang dikelola AWS saat ada fitur baru yang diluncurkan atau saat ada operasi baru yang tersedia. Layanan tidak menghapus izin yang ada di kebijakan yang dikelola AWS, sehingga pembaruan-pembaruan yang terjadi pada kebijakan tidak akan membuat izin yang ada rusak.

Selain itu, AWS mendukung kebijakan-kebijakan terkelola untuk fungsi tugas yang mencakup beberapa layanan. Misalnya, kebijakan ReadOnlyAccessAWSterkelola menyediakan akses hanya-baca ke semua AWS layanan dan sumber daya. Saat layanan meluncurkan fitur baru, AWS menambahkan izin hanya-baca untuk operasi dan sumber daya yang baru. Untuk daftar dan Deskripsi kebijakan fungsi tugas, lihat AWS kebijakan terkelola untuk fungsi tugas dalam Panduan Pengguna IAM.

Pembaruan Amazon EMR untuk kebijakan terkelola AWS

Lihat detail tentang pembaruan ke kebijakan terkelola AWS untuk Amazon EMR karena layanan ini mulai melacak perubahan ini. Untuk pemberitahuan otomatis tentang perubahan halaman ini, berlangganan ke umpan RSS pada halaman riwayat Dokumen Amazon EMR.

Perubahan	Deskripsi	Tanggal
AmazonEMRFullAccessPolicy_v2 dan **AmazonEMRReadOnlyAccessPolicy_v2** — Perbarui ke kebijakan yang ada	Ditambahkan `elasticmapreduce:ListSupportedInstanceTypes` .	13 Juli 2023
AmazonEMRFullAccessPolicy_v2 dan **AmazonEMRReadOnlyAccessPolicy_v2** — Perbarui ke kebijakan yang ada	Ditambahkan `elasticmapreduce:DescribeReleaseLabel` dan`elasticmapreduce:GetAutoTerminationPolicy` .	April 21, 2022
AmazonEMRFullAccessPolicy_v2 – Pembaruan ke kebijakan yang sudah ada	Ditambahkan `ec2:DescribeImages` untukMenggunakan AMI kustom.	Februari 15, 2022
Kebijakan terkelola Amazon EMR	Diperbarui untuk memperjelas penggunaan tag pengguna yang telah ditentukan. Menambahkan bagian tentang penggunaan AWS konsol untuk meluncurkan clsuter dengan kebijakan terkelola v2.	29 September 2021
AmazonEMRFullAccessPolicy_v2 – Pembaruan ke kebijakan yang sudah ada	Mengubah `PassRoleForAutoScaling` dan `PassRoleForEC2` tindakan untuk menggunakan operator	20 Mei 2021

Perubahan	Deskripsi	Tanggal
	`StringLike` kondisi untuk mencocokkan `"iam:PassedToService":"application-autoscaling.amazonaws.com*"` dan`"iam:PassedToService":"ec2.amazonaws.com*"` , masing-masing.	
`AmazonEMRFullAccessPolicy_v2` – Pembaruan ke kebijakan yang sudah ada	Menghapus tindakan tidak valid `s3:ListBuckets` dan diganti dengan `s3:ListAllMyBuckets` tindakan. Pembuatan peran tertaut layanan (SLR) yang diperbarui akan dicakup secara eksplisit ke satu-satunya SLR yang Amazon EMR miliki dengan utama Layanan eksplisit. SLR yang dapat dibuat adalah sama persis seperti sebelum perubahan ini.	23 Maret 2021

Perubahan	Deskripsi	Tanggal
AmazonEMRFullAccessPolicy_v2 – Kebijakan baru	Amazon EMR menambahkan izin baru untuk mencakup akses ke sumber daya dan untuk menambahkan prasyarat bahwa pengguna harus menambahkan tanda pengguna yang telah ditetapkan untuk sumber daya sebelum mereka dapat menggunakan kebijakan terkelola Amazon EMR. `iam:PassRole` tindakan memerlukan `iam:PassedToService` kondisi yang disetel ke layanan tertentu. Akses ke Amazon EC2, Amazon S3, dan layanan lainnya tidak diizinkan secara default.	11 Maret 2021
AmazonEMRServicePolicy_v2 – Kebijakan baru	Menambahkan prasyarat bahwa pengguna harus menambahkan tanda pengguna ke sumber daya sebelum mereka dapat menggunakan kebijakan ini.	11 Maret 2021
AmazonEMRReadOnlyAccessPolicy_v2 – Kebijakan baru	Izin hanya mengizinkan tindakan hanya-baca ElasticMapReduce yang ditentukan. Akses ke Amazon S3 adalah akses yang tidak diizinkan secara default.	11 Maret 2021

Perubahan	Deskripsi	Tanggal
Amazon EMR mulai melacak perubahan	Amazon EMR mulai melacak perubahan untuk kebijakan terkelola AWS.	11 Maret 2021

Kebijakan IAM untuk akses berbasis tanda ke klaster dan EMR Notebooks

Anda dapat menggunakan syarat di kebijakan berbasis identitas Anda untuk mengontrol akses ke klaster dan EMR Notebooks berdasarkan tanda.

Untuk informasi lebih lanjut tentang penambahan tanda ke klaster, lihat Klaster EMR penandaan.

Contoh berikut menunjukkan skenario yang berbeda dan cara untuk menggunakan operator syarat dengan kunci syarat Amazon EMR. Pernyataan kebijakan IAM ini dimaksudkan untuk tujuan demonstrasi saja dan tidak boleh digunakan di lingkungan produksi. Ada beberapa cara untuk menggabungkan pernyataan kebijakan untuk memberikan dan menolak izin sesuai dengan kebutuhan Anda. Untuk informasi selengkapnya tentang perencanaan dan pengujian kebijakan IAM, lihat Panduan Pengguna IAM.

> **Important**
>
> Secara eksplisit menolak izin untuk tindakan penandaan adalah pertimbangan penting. Hal ini mencegah pengguna dari penandaan sumber daya dan dengan demikian memberikan sendiri izin yang tidak ingin Anda berikan. Jika Anda tidak menolak tindakan penandaan untuk sumber daya, pengguna dapat memodifikasi tag dan menghindari maksud kebijakan berbasis tag.

Contoh pernyataan kebijakan berbasis identitas untuk klaster

Contoh berikut menunjukkan kebijakan izin berbasis identitas yang digunakan untuk mengontrol tindakan yang diizinkan dengan klaster EMR.

> **Important**
>
> Tindakan `ModifyInstanceGroup` di Amazon EMR tidak mengharuskan Anda menentukan ID klaster. Untuk alasan itu, menolak tindakan ini berdasarkan tanda klaster memerlukan

pertimbangan tambahan. Untuk informasi selengkapnya, lihat Menyangkal tindakan ModifyInstanceGroup .

Topik

- Izinkan tindakan hanya pada klaster dengan nilai tanda tertentu
- Memerlukan penandaan klaster ketika sebuah klaster dibuat
- Izinkan tindakan pada klaster dengan tanda tertentu, terlepas dari nilai tanda

Izinkan tindakan hanya pada klaster dengan nilai tanda tertentu

Contoh berikut menunjukkan kebijakan yang mengizinkan pengguna melakukan tindakan berdasarkan tanda klaster *department* dengan nilai *dev* dan juga mengizinkan pengguna untuk memberi tanda klaster dengan tanda yang sama. Kebijakan contoh akhir menunjukkan cara menolak keistimewaan untuk memberi tanda pada klaster EMR dengan apa saja tetapi harus dengan tanda yang sama.

Di kebijakan contoh berikut, syarat operator `StringEquals` mencoba untuk mencocokkan *dev* dengan nilai untuk tanda *department*. Jika tanda *department* belum ditambahkan ke klaster, atau tidak mengandung nilai *dev*, kebijakan tersebut tidak berlaku, dan tindakan tersebut tidak diizinkan oleh kebijakan ini. Jika tidak ada pernyataan kebijakan lain mengizinkan tindakan, pengguna hanya dapat bekerja dengan klaster yang memiliki tanda ini dengan nilai ini.

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Sid": "Stmt12345678901234",
      "Effect": "Allow",
      "Action": [
        "elasticmapreduce:DescribeCluster",
        "elasticmapreduce:ListSteps",
        "elasticmapreduce:TerminateJobFlows",
        "elasticmapreduce:SetTerminationProtection",
        "elasticmapreduce:ListInstances",
        "elasticmapreduce:ListInstanceGroups",
        "elasticmapreduce:ListBootstrapActions",
        "elasticmapreduce:DescribeStep"
      ],
```

```
      "Resource": [
        "*"
      ],
      "Condition": {
        "StringEquals": {
          "elasticmapreduce:ResourceTag/department": "dev"
        }
      }
    }
  ]
}
```

Anda juga dapat menentukan beberapa nilai tanda menggunakan operator syarat. Misalnya, untuk mengizinkan semua tindakan pada grup di mana tanda *department* berisi nilai *dev* atau *test*, Anda bisa mengganti blok syarat di contoh sebelumnya dengan berikut ini.

```
            "Condition": {
              "StringEquals": {
                "elasticmapreduce:ResourceTag/department":["dev", "test"]
              }
            }
```

Memerlukan penandaan klaster ketika sebuah klaster dibuat

Seperti pada contoh sebelumnya, contoh kebijakan berikut mencari tag pencocokan yang sama: nilai *dev* untuk *department* tag. Namun dalam contoh ini, kunci `RequestTag` kondisi menetapkan bahwa kebijakan berlaku selama pembuatan tag. Jadi, Anda harus membuat cluster dengan tag yang cocok dengan nilai yang ditentukan.

Untuk membuat cluster dengan tag, Anda juga harus memiliki izin untuk `elasticmapredue:AddTags` tindakan tersebut. Untuk pernyataan ini, kunci `elasticmapreduce:ResourceTag` kondisi memastikan bahwa IAM hanya memberikan akses ke sumber daya tag dengan nilai *dev* pada *department* tag. `Resource`Elemen ini digunakan untuk membatasi izin ini ke sumber daya cluster.

Untuk `PassRole` sumber daya, Anda harus memberikan ID AWS akun atau alias, nama peran layanan dalam `PassRoleForEMR` pernyataan, dan nama profil instance dalam `PassRoleForEC2` pernyataan. Untuk informasi selengkapnya tentang format ARN IAM, lihat ARN IAM di Panduan Pengguna IAM.

Untuk informasi selengkapnya tentang pencocokan nilai tag-key, lihat aws:RequestTag/tag-keydi Panduan Pengguna IAM.

```
{
    "Version": "2012-10-17",
    "Statement": [
      {
        "Sid": "RunJobFlowExplicitlyWithTag",
        "Effect": "Allow",
        "Action": [
          "elasticmapreduce:RunJobFlow"
        ],
        "Resource": "*",
        "Condition": {
          "StringEquals": {
            "aws:RequestTag/department": "dev"
          }
        }
      },
      {
        "Sid": "AddTagsForDevClusters",
        "Effect": "Allow",
        "Action": "elasticmapreduce:AddTags",
        "Resource": "arn:aws:elasticmapreduce:*:*:cluster/*",
        "Condition": {
          "StringEquals": {
            "elasticmapreduce:ResourceTag/department": "dev"
          }
        }
      },
      {
        "Sid": "PassRoleForEMR",
        "Effect": "Allow",
        "Action": "iam:PassRole",
        "Resource": "arn:aws:iam::AccountId:role/Role-Name-With-Path",
        "Condition": {
          "StringLike": {
            "iam:PassedToService": "elasticmapreduce.amazonaws.com*"
          }
        }
      },
      {
        "Sid": "PassRoleForEC2",
```

```
        "Effect": "Allow",
        "Action": "iam:PassRole",
        "Resource": "arn:aws:iam::AccountId:role/Role-Name-With-Path",
        "Condition": {
          "StringLike": {
            "iam:PassedToService": "ec2.amazonaws.com*"
          }
        }
      }
    ]
}
```

Izinkan tindakan pada klaster dengan tanda tertentu, terlepas dari nilai tanda

Anda juga dapat mengizinkan tindakan hanya pada klaster yang memiliki tanda tertentu, terlepas dari nilai tanda. Untuk melakukannya, Anda dapat menggunakan operator `Null`. Untuk informasi selengkapnya, lihat Operator syarat untuk memeriksa keberadaan kunci syarat di Panduan Pengguna IAM. Misalnya, untuk mengizinkan tindakan hanya pada klaster EMR yang memiliki *`department`* tanda, terlepas dari nilai yang dikandungnya, Anda bisa mengganti blok syarat di contoh sebelumnya dengan yang berikut. Operator `Null` mencari kehadiran tanda *`department`* pada klaster EMR. Jika tanda ada, pernyataan `Null` mengevaluasi ke SALAH, cocok dengan syarat yang ditentukan di pernyataan kebijakan ini, dan tindakan yang tepat diizinkan.

```
"Condition": {
  "Null": {
    "elasticmapreduce:ResourceTag/department":"false"
  }
}
```

Pernyataan kebijakan berikut mengizinkan pengguna untuk membuat klaster EMR hanya jika klaster akan memiliki tanda *`department`*, yang dapat berisi nilai apapun. Untuk `PassRole` sumber daya, Anda perlu memberikan ID AWS akun atau alias, dan nama peran layanan. Untuk informasi selengkapnya tentang format ARN IAM, lihat ARN IAM di Panduan Pengguna IAM.

Untuk informasi selengkapnya yang menentukan operator kondisi null (“false”), lihat Operator kondisi untuk memeriksa keberadaan kunci kondisi di Panduan Pengguna IAM.

```
{
    "Version": "2012-10-17",
```

```
    "Statement": [
        {
            "Sid": "CreateClusterTagNullCondition",
            "Effect": "Allow",
            "Action": [
                "elasticmapreduce:RunJobFlow"
            ],
            "Resource": [
                "*"
            ],
            "Condition": {
                "Null": {
                    "aws:RequestTag/department": "false"
                }
            }
        },
        {
            "Sid": "AddTagsNullCondition",
            "Effect": "Allow",
            "Action": "elasticmapreduce:AddTags",
            "Resource": "arn:aws:elasticmapreduce:*:*:cluster/*",
            "Condition": {
                "Null": {
                    "elasticmapreduce:ResourceTag/department": "false"
                }
            }
        },
        {
            "Sid": "PassRoleForElasticMapReduce",
            "Effect": "Allow",
            "Action": "iam:PassRole",
            "Resource": "arn:aws:iam::AccountId:role/Role-Name-With-Path",
            "Condition": {
                "StringLike": {
                    "iam:PassedToService": "elasticmapreduce.amazonaws.com*"
                }
            }
        },
        {
            "Sid": "PassRoleForEC2",
            "Effect": "Allow",
            "Action": "iam:PassRole",
            "Resource": "arn:aws:iam::AccountId:role/Role-Name-With-Path",
            "Condition": {
```

```
                "StringLike": {
                    "iam:PassedToService": "ec2.amazonaws.com*"
                }
            }
        }
    ]
}
```

Contoh pernyataan kebijakan berbasis identitas untuk EMR Notebooks

Contoh pernyataan kebijakan IAM di bagian ini menunjukkan skenario umum untuk menggunakan kunci untuk membatasi tindakan yang diizinkan menggunakan EMR Notebooks. Selama tidak ada kebijakan lain yang terkait dengan (pengguna) utama mengizinkan tindakan, kunci konteks syarat membatasi tindakan yang diizinkan seperti yang ditunjukkan.

Example — Izinkan akses hanya ke EMR Notebooks yang dibuat pengguna berdasarkan penandaan

Contoh pernyataan kebijakan berikut, ketika dilampirkan ke peran atau pengguna, memungkinkan pengguna untuk bekerja hanya dengan buku catatan yang telah mereka buat. Pernyataan kebijakan ini menggunakan tanda default yang diterapkan ketika notebook dibuat.

Dalam contoh, operator `StringEquals` kondisi mencoba mencocokkan variabel yang mewakili pengguna saat ini ID pengguna (`{aws:userId}`) dengan nilai tag`creatorUserID`. Jika tanda `creatorUserID` belum ditambahkan ke notebook, atau tidak berisi nilai ID pengguna saat ini, kebijakan tidak berlaku, dan tindakan tersebut tidak diizinkan oleh kebijakan ini. Jika tidak ada pernyataan kebijakan lain mengizinkan tindakan, pengguna hanya dapat bekerja dengan notebook yang memiliki tanda ini dengan nilai ini.

```
{
    "Version": "2012-10-17",
    "Statement": [
        {
            "Action": [
                "elasticmapreduce:DescribeEditor",
                "elasticmapreduce:StartEditor",
                "elasticmapreduce:StopEditor",
                "elasticmapreduce:DeleteEditor",
                "elasticmapreduce:OpenEditorInConsole"
            ],
            "Effect": "Allow",
            "Resource": "*",
            "Condition": {
```

```
                "StringEquals": {
                    "elasticmapreduce:ResourceTag/creatorUserId": "${aws:userId}"
                }
            }
        }
    ]
}
```

Example –Memerlukan penandaan notebook saat notebook dibuat

di contoh ini, kunci konteks `RequestTag` digunakan. Tindakan `CreateEditor` diperbolehkan hanya jika pengguna tidak mengubah atau menghapus tanda `creatorUserID` yang ditambahkan secara default. Variabel ${aws:userId}, menentukan ID Pengguna dari pengguna aktif saat ini, yang merupakan nilai default dari tanda.

Pernyataan kebijakan dapat digunakan untuk membantu memastikan bahwa pengguna tidak menghapus tanda `createUserId` atau mengubah nilainya.

```
{
    "Version": "2012-10-17",
    "Statement": [
        {
            "Action": [
                "elasticmapreduce:CreateEditor"
            ],
            "Effect": "Allow",
            "Resource": "*",
            "Condition": {
                "StringEquals": {
                    "elasticmapreduce:RequestTag/creatorUserId": "${aws:userid}"
                }
            }
        }
    ]
}
```

Contoh ini mengharuskan pengguna membuat klaster dengan tanda yang membuat string kunci `dept` dan nilai diatur ke salah satu langkah berikut: `datascience`, `analytics`, `operations`.

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

```
        {
            "Action": [
                "elasticmapreduce:CreateEditor"
            ],
            "Effect": "Allow",
            "Resource": "*",
            "Condition": {
                "StringEquals": {
                    "elasticmapreduce:RequestTag/dept": [
                        "datascience",
                        "analytics",
                        "operations"
                    ]
                }
            }
        }
    ]
}
```

Example –Batasi pembuatan notebook ke klaster yang ditandai, dan memerlukan tanda notebook

Contoh ini mengizinkan pembuatan notebook hanya jika notebook dibuat dengan tanda yang memiliki string kunci `owner` yang diatur ke salah satu nilai yang ditentukan. Selain itu, notebook hanya bisa dibuat jika klaster memiliki tanda dengan string kunci `department` yang diatur ke salah satu nilai yang ditentukan.

```
{
    "Version": "2012-10-17",
    "Statement": [
        {
            "Action": [
                "elasticmapreduce:CreateEditor"
            ],
            "Effect": "Allow",
            "Resource": "*",
            "Condition": {
                "StringEquals": {
                    "elasticmapreduce:RequestTag/owner": [
                        "owner1",
                        "owner2",
                        "owner3"
                    ],
                    "elasticmapreduce:ResourceTag/department": [
```

```
                        "dep1",
                        "dep3"
                    ]
                }
            }
        }
    ]
}
```

Example –Batasi kemampuan mulai notebook berdasarkan tanda

Contoh ini membatasi kemampuan untuk memulai notebook hanya untuk notebook yang memiliki tanda dengan string kunci `owner` yang diatur ke salah satu nilai yang ditentukan. Karena elemen `Resource` digunakan hanya untuk menentukan `editor`, syarat tidak berlaku untuk klaster, dan tidak perlu ditandai.

```
{
    "Version": "2012-10-17",
    "Statement": [
        {
            "Action": [
                "elasticmapreduce:StartEditor"
            ],
            "Effect": "Allow",
            "Resource": "arn:aws:elasticmapreduce:*:123456789012:editor/*",
            "Condition": {
                "StringEquals": {
                    "elasticmapreduce:ResourceTag/owner": [
                        "owner1",
                        "owner2"
                    ]
                }
            }
        }
    ]
}
```

Contoh ini mirip dengan yang di atas. Namun, batas hanya berlaku untuk grup yang ditandai, bukan notebook.

```
{
    "Version": "2012-10-17",
```

```
    "Statement": [
        {
            "Action": [
                "elasticmapreduce:StartEditor"
            ],
            "Effect": "Allow",
            "Resource": "arn:aws:elasticmapreduce:*:123456789012:cluster/*",
            "Condition": {
                "StringEquals": {
                    "elasticmapreduce:ResourceTag/department": [
                        "dep1",
                        "dep3"
                    ]
                }
            }
        }
    ]
}
```

Contoh ini menggunakan seperangkat notebook dan tanda klaster yang berbeda. Hal ini mengizinkan notebook untuk dimulai hanya jika:

- Notebook ini memiliki tanda dengan string kunci `owner` yang diatur ke salah satu nilai yang ditentukan

  —dan—

- Klaster memiliki tanda dengan string kunci `department` yang diatur ke salah satu nilai yang ditentukan

```
{
    "Version": "2012-10-17",
    "Statement": [
        {
            "Action": [
                "elasticmapreduce:StartEditor"
            ],
            "Effect": "Allow",
            "Resource": "arn:aws:elasticmapreduce:*:123456789012:editor/*",
            "Condition": {
                "StringEquals": {
                    "elasticmapreduce:ResourceTag/owner": [
```

```
                        "user1",
                        "user2"
                    ]
                }
            }
        },
        {
            "Action": [
                "elasticmapreduce:StartEditor"
            ],
            "Effect": "Allow",
            "Resource": "arn:aws:elasticmapreduce:*:123456789012:cluster/*",
            "Condition": {
                "StringEquals": {
                    "elasticmapreduce:ResourceTag/department": [
                        "datascience",
                        "analytics"
                    ]
                }
            }
        }
    ]
}
```

Example –Batasi kemampuan untuk membuka editor notebook berdasarkan tanda

Contoh ini mengizinkan editor notebook dibuka hanya jika:

- Notebook ini memiliki tanda dengan string kunci `owner` yang diatur ke salah satu nilai yang ditentukan.

  —dan—

- Klaster memiliki tanda dengan string kunci `department` yang diatur ke salah satu nilai yang ditentukan.

```
{
    "Version": "2012-10-17",
    "Statement": [
        {
            "Action": [
                "elasticmapreduce:OpenEditorInConsole"
            ],
```

```
            "Effect": "Allow",
            "Resource": "arn:aws:elasticmapreduce:*:123456789012:editor/*",
            "Condition": {
                "StringEquals": {
                    "elasticmapreduce:ResourceTag/owner": [
                        "user1",
                        "user2"
                    ]
                }
            }
        },
        {
            "Action": [
                "elasticmapreduce:OpenEditorInConsole"
            ],
            "Effect": "Allow",
            "Resource": "arn:aws:elasticmapreduce:*:123456789012:cluster/*",
            "Condition": {
                "StringEquals": {
                    "elasticmapreduce:ResourceTag/department": [
                        "datascience",
                        "analytics"
                    ]
                }
            }
        }
    ]
}
```

Menyangkal tindakan ModifyInstanceGroup

ModifyInstanceGroupsTindakan di Amazon EMR tidak mengharuskan Anda memberikan ID cluster dengan tindakan tersebut. Sebaliknya, Anda dapat hanya menentukan ID grup instans. Untuk alasan ini, langsung tolak kebijakan untuk tindakan ini berdasarkan ID klaster atau tanda klaster mungkin tidak memiliki efek yang dimaksudkan. Pertimbangkan kebijakan contoh berikut.

```
{
 "Version": "2012-10-17",
    "Statement": [
        {
            "Action": [
                "elasticmapreduce:ModifyInstanceGroups"
            ],
```

```
            "Effect": "Allow",
            "Resource": "*"
        },
        {
            "Action": [
                "elasticmapreduce:ModifyInstanceGroups"
            ],
            "Effect": "Deny",
            "Resource": "arn:aws:elasticmapreduce:us-east-1:123456789012:cluster/
j-12345ABCDEFG67"
        }
    ]

}
```

Jika pengguna dengan kebijakan terlampir ini melakukan tindakan `ModifyInstanceGroup` dan menentukan hanya ID grup instans, kebijakan tidak berlaku. Karena tindakan diizinkan pada semua sumber daya lainnya, tindakan tersebut berhasil.

Solusi untuk masalah ini adalah melampirkan pernyataan kebijakan ke identitas yang menggunakan NotResourceelemen untuk menolak `ModifyInstanceGroup` tindakan apa pun yang dikeluarkan tanpa ID klaster. Kebijakan contoh berikut menambahkan pernyataan tolak sehingga setiap permintaan `ModifyInstanceGroups` gagal kecuali ID klaster ditentukan. Karena identitas harus menentukan ID klaster dengan tindakan, tolak pernyataan berdasarkan ID klaster karena efektif.

```
{
 "Version": "2012-10-17",
    "Statement": [
        {
            "Action": [
                "elasticmapreduce:ModifyInstanceGroups"
            ],
            "Effect": "Allow",
            "Resource": "*"
        },
        {
            "Action": [
                "elasticmapreduce:ModifyInstanceGroups"
            ],
            "Effect": "Deny",
            "Resource": "arn:aws:elasticmapreduce:us-east-1:123456789012:cluster/
j-12345ABCDEFG67"
```

```
        },
        {
            "Action": [
                "elasticmapreduce:ModifyInstanceGroups"
            ],
            "Effect": "Deny",
            "NotResource": "arn:*:elasticmapreduce:*:*:cluster/*"
        }
    ]

}
```

Masalah serupa ada saat Anda ingin tolak tindakan `ModifyInstanceGroups` berdasarkan nilai yang terkait dengan tanda klaster. Solusinya serupa. Selain tolak pernyataan yang menentukan nilai tanda, Anda dapat menambahkan pernyataan kebijakan yang menolak tindakan `ModifyInstanceGroup` tanda yang Anda tentukan tidak ada, terlepas dari nilai.

Contoh berikut menunjukkan kebijakan yang, saat dilampirkan ke identitas, menolak identitas tindakan `ModifyInstanceGroups` klaster apapun dengan tanda `department` yang diatur ke `dev`. Pernyataan ini hanya efektif karena pernyataan tolak menggunakan syarat `StringNotLike` untuk menolak tindakan kecuali tanda `department` ada.

```
{
    "Version": "2012-10-17",
    "Statement": [
        {
            "Action": [
                "elasticmapreduce:ModifyInstanceGroups"
            ],
            "Effect": "Allow",
            "Resource": "*"
        },
        {
            "Action": [
                "elasticmapreduce:ModifyInstanceGroups"
            ],
            "Condition": {
                "StringEquals": {
                    "aws:ResourceTag/department": "dev"
                }
            },
            "Effect": "Deny",
```

```
            "Resource": "*"
        },
        {
            "Action": [
                "elasticmapreduce:ModifyInstanceGroups"
            ],
            "Condition": {
                "StringNotLike": {
                    "aws:ResourceTag/department": "?*"
                }
            },
            "Effect": "Deny",
            "Resource": "*"
        }
    ],
}
```

Memecahkan masalah identitas dan akses EMR Amazon

Gunakan informasi berikut untuk membantu Anda mendiagnosis dan memperbaiki masalah umum yang mungkin Anda temui saat bekerja dengan Amazon EMR dan IAM.

Topik

- Saya tidak berwenang untuk melakukan tindakan di Amazon EMR
- Saya tidak berwenang untuk melakukan iam: PassRole
- Saya ingin mengizinkan orang di luar AWS akun saya untuk mengakses sumber daya EMR Amazon saya

Saya tidak berwenang untuk melakukan tindakan di Amazon EMR

Jika AWS Management Console memberi tahu bahwa Anda tidak diotorisasi untuk melakukan tindakan, Anda harus menghubungi administrator untuk mendapatkan bantuan. Administrator adalah orang yang memberikan nama pengguna dan kata sandi Anda untuk Anda.

Contoh kesalahan berikut terjadi ketika `mateojackson` pengguna mencoba menggunakan konsol untuk melihat detail tentang *`my-example-widget`* sumber daya fiksi tetapi tidak memiliki izin EMR:*`GetWidget`* fiksi.

```
User: arn:aws:iam::123456789012:user/mateojackson is not authorized to perform:
 EMR:GetWidget on resource: my-example-widget
```

Dalam hal ini, Mateo meminta administratornya untuk memperbarui kebijakannya untuk memungkinkannya mengakses sumber daya *my-example-widget* dengan menggunakan tindakan EMR:*GetWidget*.

Saya tidak berwenang untuk melakukan iam: PassRole

Jika Anda menerima kesalahan bahwa Anda tidak diizinkan untuk melakukan `iam:PassRole` tindakan, kebijakan Anda harus diperbarui agar Anda dapat meneruskan peran ke Amazon EMR.

Sebagian Layanan AWS mengizinkan Anda untuk memberikan peran yang sudah ada ke layanan tersebut alih-alih membuat peran layanan baru atau peran tertaut-layanan. Untuk melakukan tindakan tersebut, Anda harus memiliki izin untuk memberikan peran pada layanan tersebut.

Contoh kesalahan berikut terjadi ketika pengguna IAM bernama `marymajor` mencoba menggunakan konsol untuk melakukan tindakan di Amazon EMR. Namun, tindakan tersebut memerlukan layanan untuk mendapatkan izin yang diberikan oleh peran layanan. Mary tidak memiliki izin untuk meneruskan peran tersebut pada layanan.

```
User: arn:aws:iam::123456789012:user/marymajor is not authorized to perform:
 iam:PassRole
```

Dalam kasus ini, kebijakan Mary harus diperbarui agar dia mendapatkan izin untuk melakukan tindakan `iam:PassRole` tersebut.

Jika Anda membutuhkan bantuan, hubungi administrator AWS Anda. Administrator Anda adalah orang yang memberikan kredensial masuk Anda.

Saya ingin mengizinkan orang di luar AWS akun saya untuk mengakses sumber daya EMR Amazon saya

Anda dapat membuat peran yang dapat digunakan para pengguna di akun lain atau orang-orang di luar organisasi Anda untuk mengakses sumber daya Anda. Anda dapat menentukan siapa yang dipercaya untuk mengambil peran tersebut. Untuk layanan yang mendukung kebijakan berbasis sumber daya atau daftar kontrol akses (ACL), Anda dapat menggunakan kebijakan tersebut untuk memberi akses kepada orang ke sumber daya Anda.

Untuk mempelajari selengkapnya, periksa hal berikut:

- Untuk mengetahui apakah Amazon EMR mendukung fitur-fitur ini, lihat. Cara kerja Amazon EMR dengan IAM

- Untuk mempelajari cara memberikan akses ke sumber daya di seluruh Akun AWS yang Anda miliki, silakan lihat Menyediakan akses ke pengguna IAM di Akun AWS lainnya yang Anda miliki di Panduan Pengguna IAM.
- Untuk mempelajari cara memberikan akses ke sumber daya Anda ke pihak ketiga Akun AWS, silakan lihat Menyediakan akses ke akun Akun AWS yang dimiliki oleh pihak ketiga dalam Panduan Pengguna IAM.
- Untuk mempelajari cara memberikan akses melalui federasi identitas, silakan lihat Menyediakan akses ke pengguna terautentikasi eksternal (gabungan identitas) di Panduan Pengguna IAM .
- Untuk mempelajari perbedaan antara penggunaan kebijakan peran dan kebijakan berbasis sumber daya untuk akses lintas akun, silakan lihat Bagaimana peran IAM berbeda dari kebijakan berbasis sumber daya di Panduan Pengguna IAM.

Menggunakan Hibah Akses Amazon S3 dengan Amazon EMR

Ikhtisar Hibah Akses S3 untuk Amazon EMR

Dengan Amazon EMR rilis 6.15.0 dan yang lebih tinggi, Amazon S3 Access Grants menyediakan solusi kontrol akses yang dapat diskalakan yang dapat Anda gunakan untuk menambah akses ke data Amazon S3 Anda dari Amazon EMR. Jika Anda memiliki konfigurasi izin yang kompleks atau besar untuk data S3, Anda dapat menggunakan Access Grants untuk menskalakan izin data S3 untuk pengguna, peran, dan aplikasi di klaster Anda.

Gunakan S3 Access Grants untuk menambah akses ke data Amazon S3 di luar izin yang diberikan oleh peran runtime atau peran IAM yang dilampirkan ke identitas dengan akses ke cluster EMR Anda. Untuk informasi selengkapnya, lihat Mengelola akses dengan Hibah Akses S3 di Panduan Pengguna Amazon S3.

Untuk langkah-langkah menggunakan Hibah Akses S3 dengan penerapan EMR Amazon lainnya, lihat dokumentasi berikut:

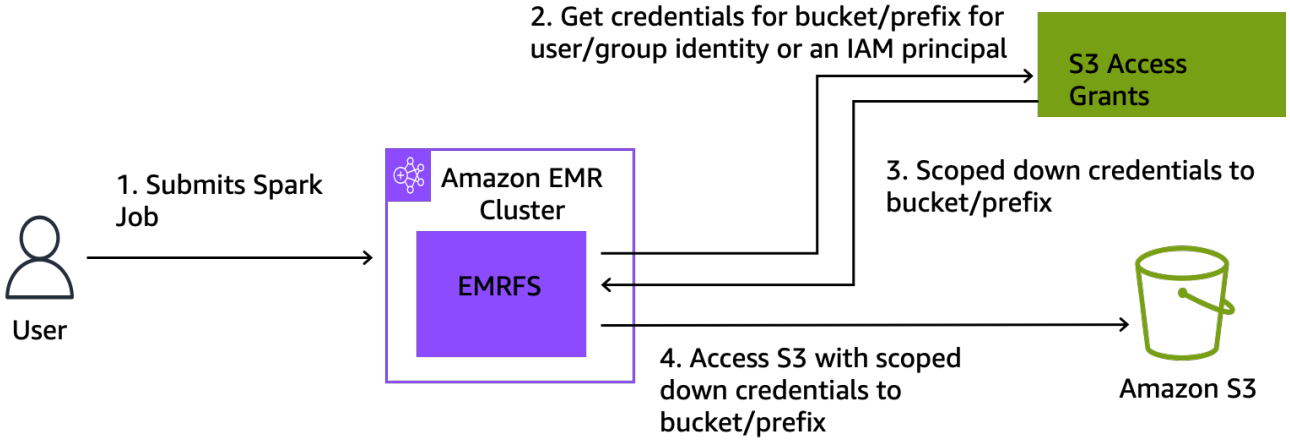
- Menggunakan Hibah Akses S3 dengan Amazon EMR di EKS
- Menggunakan Hibah Akses S3 dengan Amazon EMR Tanpa Server

Bagaimana Amazon EMR bekerja dengan S3 Access Grants

Amazon EMR merilis 6.15.0 dan yang lebih tinggi menyediakan integrasi asli dengan S3 Access Grants. Anda dapat mengaktifkan S3 Access Grants di Amazon EMR dan menjalankan pekerjaan

Spark. Saat pekerjaan Spark membuat permintaan untuk data S3, Amazon S3 memberikan kredensyal sementara yang dicakup ke bucket, awalan, atau objek tertentu.

Berikut ini adalah ikhtisar tingkat tinggi tentang bagaimana Amazon EMR mendapatkan akses ke data yang dilindungi oleh S3 Access Grants.



1. Seorang pengguna mengirimkan pekerjaan Amazon EMR Spark yang menggunakan data yang disimpan di Amazon S3.
2. Amazon EMR membuat permintaan S3 Access Grants untuk mengizinkan akses ke bucket, awalan, atau objek atas nama pengguna tersebut.
3. Amazon S3 mengembalikan kredensyal sementara dalam bentuk token AWS Security Token Service (STS) untuk pengguna. Token dicakup untuk mengakses bucket, awalan, atau objek S3.
4. Amazon EMR menggunakan token STS untuk mengambil data dari S3.
5. Amazon EMR menerima data dari S3 dan mengembalikan hasilnya kepada pengguna.

Akses S3 Memberikan pertimbangan dengan Amazon EMR

Perhatikan perilaku dan batasan berikut saat Anda menggunakan S3 Access Grants dengan Amazon EMR.

Dukungan fitur

- S3 Access Grants didukung dengan Amazon EMR rilis 6.15.0 dan yang lebih tinggi.
- Spark adalah satu-satunya mesin kueri yang didukung saat Anda menggunakan S3 Access Grants dengan Amazon EMR.

- Delta Lake dan Hudi adalah satu-satunya format meja terbuka yang didukung saat Anda menggunakan Hibah Akses S3 dengan Amazon EMR.
- Kemampuan EMR Amazon berikut tidak didukung untuk digunakan dengan Hibah Akses S3:
  - Tabel Apache Iceberg
  - Otentikasi asli LDAP
  - Autentikasi asli Apache Ranger
  - AWS CLIpermintaan ke Amazon S3 yang menggunakan peran IAM
  - Akses S3 melalui protokol sumber terbuka S3A
- `fallbackToIAM`Opsi ini tidak didukung untuk kluster EMR yang menggunakan propagasi identitas tepercaya dengan IAM Identity Center.
- Hibah Akses S3 dengan hanya AWS Lake Formation didukung dengan kluster EMR Amazon yang berjalan di Amazon EC2.

Pertimbangan perilaku

- Integrasi asli Apache Ranger dengan Amazon EMR memiliki fungsionalitas yang kongruen dengan S3 Access Grants sebagai bagian dari plugin EMRFS S3 Apache Ranger. Jika Anda menggunakan Apache Ranger untuk kontrol akses halus (FGAC), kami sarankan Anda menggunakan plugin itu alih-alih S3 Access Grants.
- Amazon EMR menyediakan cache kredensyal di EMRFS untuk memastikan bahwa pengguna tidak perlu membuat permintaan berulang untuk kredensyal yang sama dalam pekerjaan Spark. Oleh karena itu, Amazon EMR selalu meminta hak istimewa tingkat default saat meminta kredensyal. Untuk informasi selengkapnya, lihat Meminta akses ke data S3 di Panduan Pengguna Amazon S3.
- Jika pengguna melakukan tindakan yang tidak didukung oleh S3 Access Grants, Amazon EMR disetel untuk menggunakan peran IAM yang ditentukan untuk eksekusi pekerjaan. Untuk informasi selengkapnya, lihat Kembali ke peran IAM.

Luncurkan klaster EMR Amazon dengan Hibah Akses S3

Bagian ini menjelaskan cara meluncurkan klaster EMR yang berjalan di Amazon EC2, dan menggunakan Hibah Akses S3 untuk mengelola akses ke data di Amazon S3. Untuk langkah-langkah menggunakan Hibah Akses S3 dengan penerapan EMR Amazon lainnya, lihat dokumentasi berikut:

- Menggunakan Hibah Akses S3 dengan Amazon EMR di EKS
- Menggunakan Hibah Akses S3 dengan EMR Tanpa Server

Gunakan langkah-langkah berikut untuk meluncurkan klaster EMR yang berjalan di Amazon EC2, dan menggunakan S3 Access Grants untuk mengelola akses ke data di Amazon S3.

1. Siapkan peran eksekusi pekerjaan untuk klaster EMR Anda. Sertakan izin IAM yang diperlukan yang Anda perlukan untuk menjalankan pekerjaan Spark, dan: `s3:GetDataAccess` `s3:GetAccessGrantsInstanceForPrefix`

```
{
    "Effect": "Allow",
    "Action": [
    "s3:GetDataAccess",
    "s3:GetAccessGrantsInstanceForPrefix"
    ],
    "Resource": [     //LIST ALL INSTANCE ARNS THAT THE ROLE IS ALLOWED TO QUERY
         "arn:aws_partition:s3:Region:account-id1:access-grants/default",
         "arn:aws_partition:s3:Region:account-id2:access-grants/default"
    ]
}
```

> **Note**
>
> Dengan Amazon EMR, S3 Access Grants menambah izin yang ditetapkan dalam peran IAM. Jika peran IAM yang Anda tentukan untuk eksekusi pekerjaan berisi izin untuk mengakses S3 secara langsung, maka pengguna mungkin dapat mengakses lebih banyak data daripada hanya data yang Anda tentukan di S3 Access Grants.

2. Selanjutnya, gunakan AWS CLI untuk membuat klaster dengan Amazon EMR 6.15 atau lebih tinggi dan `emrfs-site` klasifikasi untuk mengaktifkan S3 Access Grants, mirip dengan contoh berikut:

```
aws emr create-cluster
  --release-label emr-6.15.0 \
  --instance-count 3 \
  --instance-type m5.xlarge \
```

```
 --configurations '[{"Classification":"emrfs-site",
"Properties":{"fs.s3.s3AccessGrants.enabled":"true",
"fs.s3.s3AccessGrants.fallbackToIAM":"false"}}]'
```

Hibah Akses S3 dengan AWS Lake Formation

Jika Anda menggunakan Amazon EMR dengan AWS Lake Formationintegrasi, Anda dapat menggunakan Amazon S3 Access Grants untuk akses langsung atau tabular ke data di Amazon S3.

> **Note**
>
> Hibah Akses S3 dengan hanya AWS Lake Formation didukung dengan kluster EMR Amazon yang berjalan di Amazon EC2.

Akses langsung

Akses langsung melibatkan semua panggilan untuk mengakses data S3 yang tidak memanggil API untuk layanan AWS Glue yang digunakan Lake Formation sebagai metastore dengan Amazon EMR, misalnya, untuk memanggil: `spark.read`

```
spark.read.csv("s3://...")
```

Saat Anda menggunakan Hibah Akses S3 dengan EMR AWS Lake Formation Amazon, semua pola akses langsung melalui Hibah Akses S3 untuk mendapatkan kredensyal S3 sementara.

Akses tabular

Akses tabular terjadi saat Lake Formation memanggil API metastore untuk mengakses lokasi S3 Anda, misalnya, untuk menanyakan data tabel:

```
spark.sql("select * from test_tbl")
```

Saat Anda menggunakan S3 Access Grants dengan EMR AWS Lake Formation Amazon, semua pola akses tabular melewati Lake Formation.

Kembali ke peran IAM

Jika pengguna mencoba melakukan tindakan yang tidak didukung oleh S3 Access Grants, Amazon EMR akan default ke peran IAM yang ditentukan untuk eksekusi pekerjaan saat konfigurasi dilakukan. `fallbackToIAM` `true` Hal ini memungkinkan pengguna untuk kembali pada peran eksekusi pekerjaan mereka untuk memberikan kredensyal untuk akses S3 dalam skenario yang tidak dicakup oleh S3 Access Grants.

Dengan `fallbackToIAM` diaktifkan, pengguna dapat mengakses data yang diizinkan oleh Access Grant. Jika tidak ada token Hibah Akses S3 untuk data target, maka Amazon EMR memeriksa izin pada peran eksekusi pekerjaan mereka.

> **Note**
>
> Kami menyarankan Anda menguji izin akses Anda dengan `fallbackToIAM` konfigurasi diaktifkan bahkan jika Anda berencana untuk menonaktifkan opsi untuk beban kerja produksi. Dengan pekerjaan Spark, ada cara lain agar pengguna dapat mengakses semua set izin dengan kredensyal IAM mereka. Saat diaktifkan pada kluster EMR, hibah dari S3 memberikan akses pekerjaan Spark ke lokasi S3. Anda harus memastikan bahwa Anda melindungi lokasi S3 ini dari akses di luar EMRFS. Misalnya, Anda harus melindungi lokasi S3 dari akses oleh klien S3 yang digunakan di notebook, atau oleh aplikasi yang tidak didukung oleh S3 Access Grants seperti Hive atau Presto.

Autentikasi ke simpul klaster Amazon EMR

Klien SSH dapat menggunakan pasangan kunci Amazon EC2 untuk mengautentikasi ke instans klaster. Atau, dengan Amazon EMR rilis 5.10.0 dan yang lebih tinggi, Anda dapat mengonfigurasi Kerberos untuk mengautentikasi pengguna dan koneksi SSH ke node utama. Dan dengan Amazon EMR rilis 5.12.0 dan lebih tinggi, Anda dapat mengautentikasi dengan LDAP.

Topik

- Menggunakan key pair EC2 untuk kredensi SSH
- Gunakan Kerberos untuk otentikasi dengan Amazon EMR
- Gunakan Active Directory atau server LDAP untuk otentikasi dengan Amazon EMR

Menggunakan key pair EC2 untuk kredensi SSH

Simpul klaster Amazon EMR berjalan di instans Amazon EC2. Anda dapat connect ke simpul klaster dengan cara yang sama seperti Anda dapat connect ke instans Amazon EC2. Anda dapat menggunakan Amazon EC2 untuk menciptakan sebuah pasangan kunci baru, atau Anda dapat mengimpor sebuah pasangan kunci yang sudah ada. Ketika Anda membuat sebuah klaster, Anda dapat menentukan psangan kunci Amazon EC2 yang akan digunakan untuk koneksi SSH untuk semua instans klaster. Anda juga dapat membuat klaster tanpa psangan kunci. Hal ini biasanya dilakukan dengan klaster sementara yang mulai, menjalankan langkah-langkah, dan versi terbaru mengakhiri secara otomatis.

Klien SSH yang Anda gunakan connect ke klaster perlu menggunakan file kunci privat yang terkait dengan pasangan kunci ini. Ini adalah file .pem untuk klien SSH yang menggunakan Linux, Unix dan macOS. Anda harus mengatur izin sehingga hanya pemilik kunci yang memiliki izin untuk mengakses file. Ini adalah file .ppk untuk klien SSH menggunakan Windows, dan file .ppk biasanya dibuat dari file .pem.

- Untuk informasi lebih lanjut tentang membuat pasangan kunci Amazon EC2, lihat pasangan kunci Amazon EC2 di Panduan Pengguna Amazon EC2 untuk Instans Linux.
- Untuk petunjuk tentang menggunakan PuTTYgen untuk membuat file .ppk dari file .pem, lihat Mengubah kunci privat Anda menggunakan PuTTYgen di Amazon EC2 untuk Instans Linux.
- Untuk informasi selengkapnya tentang menyetel izin file.pem dan cara menyambung ke simpul utama klaster EMR menggunakan metode yang berbeda - termasuk dari `ssh` Linux atau macOS, Putty dari Windows, atau dari sistem operasi apa pun yang didukung, lihat. AWS CLI Connect ke node utama menggunakan SSH

Gunakan Kerberos untuk otentikasi dengan Amazon EMR

Amazon EMR merilis 5.10.0 dan dukungan yang lebih tinggi Kerberos. Kerberos adalah protokol otentikasi jaringan yang menggunakan kriptografi kunci rahasia untuk memberikan otentikasi yang kuat sehingga kata sandi atau kredensyal lainnya tidak dikirim melalui jaringan dalam format yang tidak terenkripsi.

Di Kerberos, layanan dan pengguna yang perlu mengautentikasi dikenal sebagai utama. Utama ada di ranah Kerberos. Di ranah, server Kerberos yang dikenal sebagai pusat distribusi kunci (KDC) menyediakan sarana bagi utama untuk mengautentikasi. KDC melakukan ini dengan mengeluarkan tiket untuk autentikasi. KDC mempertahankan basis data utama dari ranah, kata sandi mereka, dan

informasi administratif lainnya tentang setiap utama. KDC juga dapat menerima kredensial autentikasi dari utama di ranah lain, yang dikenal sebagai kepercayaan lintas ranah. Selain itu, klaster EMR dapat menggunakan KDC eksternal untuk mengautentikasi utama.

Skenario umum untuk membangun kepercayaan lintas ranah atau menggunakan KDC eksternal adalah untuk mengautentikasi pengguna dari domain Direktori Aktif. Hal ini memungkinkan pengguna untuk mengakses kluster EMR dengan akun domain mereka ketika mereka menggunakan SSH untuk terhubung ke cluster atau bekerja dengan aplikasi data besar.

Ketika Anda menggunakan autentikasi Kerberos, Amazon EMR mengonfigurasi Kerberos untuk aplikasi, komponen, dan subsistem yang diinstal di klaster sehingga mereka saling beratutentikasi satu sama lain.

> **Important**
>
> Amazon EMR tidak mendukung AWS Directory Service for Microsoft Active Directory di kepercayaan lintas ranah atau sebagai KDC eksternal.

Sebelum Anda mengonfigurasi Kerberos menggunakan Amazon EMR, kami merekomendasikan Anda agar familiar dengan konsep Kerberos, layanan yang berjalan pada KDC, dan alat-alat untuk mengelola layanan Kerberos. Untuk informasi selengkapnya, lihat Dokumentasi Kerberos MIT, yang diterbitkan oleh Konsorsium Kerberos.

Topik

Aplikasi-aplikasi yang didukung

Dalam klaster EMR, utama Kerberos adalah layanan aplikasi big data dan subsistem yang berjalan pada semua simpul klaster. Amazon EMR dapat mengonfigurasi aplikasi dan komponen yang

tercantum di bawah ini untuk menggunakan Kerberos. Setiap aplikasi memiliki utama pengguna Kerberos yang terkait dengannya.

Amazon EMR tidak support kepercayaan lintas ranah dengan AWS Directory Service for Microsoft Active Directory.

Amazon EMR hanya mengonfigurasi fitur autentikasi Kerberos sumber daya terbuka untuk aplikasi dan komponen yang tercantum di bawah ini. Aplikasi lain yang diinstal bukan Kerberized, yang dapat mengakibatkan ketidakmampuan untuk berkomunikasi dengan komponen Kerberized dan menyebabkan kesalahan aplikasi. Aplikasi dan komponen yang bukan Kerberized tidak mengaktifkan autentikasi. Aplikasi dan komponen yang didukung dapat bervariasi untuk rilis EMR Amazon yang berbeda.

Antarmuka pengguna Livy adalah satu-satunya antarmuka pengguna web yang dihosting di cluster yang Kerberized.

- Hadoop MapReduce
- Hbase
- HCatalog
- HDFS
- Sarang
  - Jangan mengaktifkan Hive dengan autentikasi LDAP. Hal ini dapat menyebabkan masalah komunikasi dengan YARN Kerberized.
- Rona
  - Autentikasi pengguna Hue tidak diatur secara otomatis dan dapat dikonfigurasi menggunakan API konfigurasi.
  - Server Hue adalah Kerberized. Hue front-end (UI) tidak dikonfigurasi untuk autentikasi. Autentikasi LDAP dapat dikonfigurasi untuk UI Hue.
- Livy
  - Peniruan identitas dengan cluster Kerberized didukung di Amazon EMR rilis 5.22.0 dan yang lebih tinggi.
- Oozie
- Phoenix
- Presto
  - Presto mendukung otentikasi Kerberos di Amazon EMR rilis 6.9.0 dan lebih tinggi.

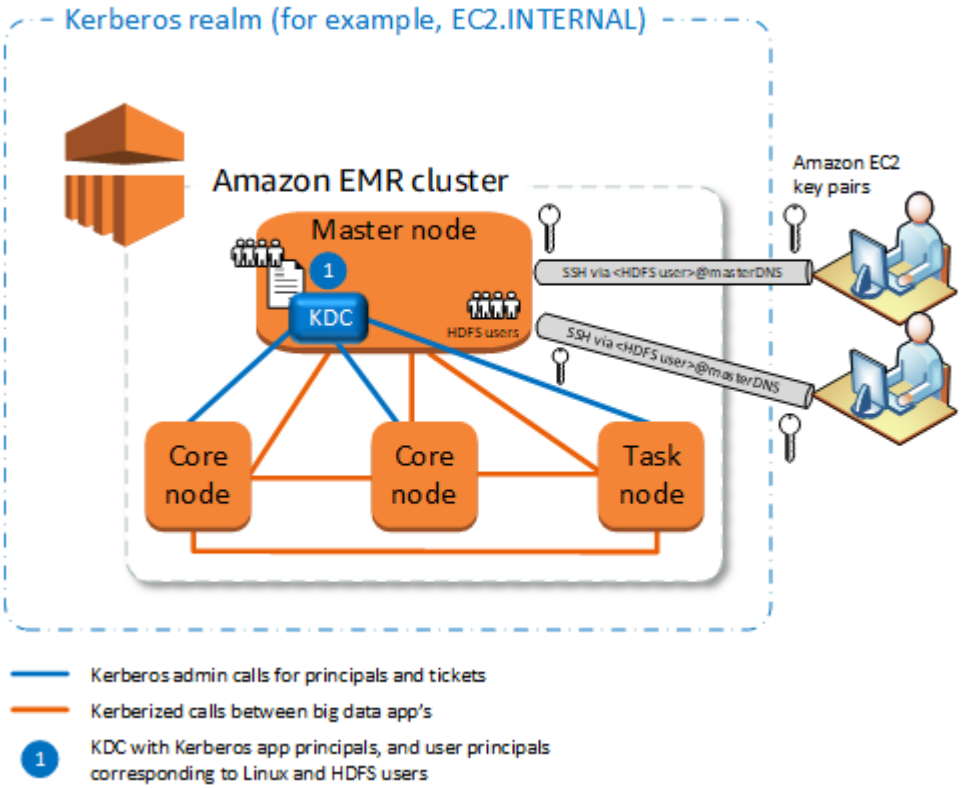
  - Untuk menggunakan otentikasi Kerberos untuk Presto, Anda harus mengaktifkan enkripsi dalam transit.
- Percikan
- Tez
- Trino
  - Trino mendukung otentikasi Kerberos di Amazon EMR rilis 6.11.0 dan lebih tinggi.
  - Untuk menggunakan otentikasi Kerberos untuk Trino, Anda harus mengaktifkan enkripsi dalam transit.
- BENANG
- Zeppelin
  - Zeppelin hanya dikonfigurasi untuk menggunakan Kerberos dengan interpreter Spark. Zeppelin ini tidak dikonfigurasi untuk penerjemah lain.
  - Peniruan identitas pengguna tidak didukung untuk penerjemah Zeppelin Kerberized selain Spark.
- Penjaga kebun binatang
  - Zookeeper klien tidak didukung.

Pilihan arsitektur Kerberos

Bila Anda menggunakan Kerberos dengan Amazon EMR, Anda dapat memilih dari arsitektur yang tercantum di bagian ini. Terlepas dari arsitektur yang Anda pilih, Anda mengonfigurasi Kerberos menggunakan langkah yang sama. Anda membuat konfigurasi keamanan, Anda menentukan konfigurasi keamanan dan opsi Kerberos spesifik klaster yang kompatibel, dan Anda membuat direktori HDFS untuk pengguna Linux di klaster yang sesuai dengan utama pengguna di KDC. Untuk penjelasan tentang opsi konfigurasi dan konfigurasi contoh untuk setiap arsitektur, lihat Mengonfigurasi Kerberos di Amazon EMR.

KDC khusus cluster (KDC pada simpul utama)

Konfigurasi ini tersedia dengan Amazon EMR rilis 5.10.0 dan lebih tinggi.



Keuntungan

- Amazon EMR memiliki kepemilikan penuh KDC.
- KDC pada klaster EMR independen dari implementasi KDC terpusat seperti Direktori Aktif Microsoft atau AWS Managed Microsoft AD.
- Dampak performa minimal karena KDC mengelola autentikasi hanya untuk simpul lokal di klaster.
- Opsional, klaster Kerberized lainnya dapat referensi KDC sebagai KDC eksternal. Untuk informasi selengkapnya, lihat KDC eksternal — Node primer pada cluster yang berbeda.

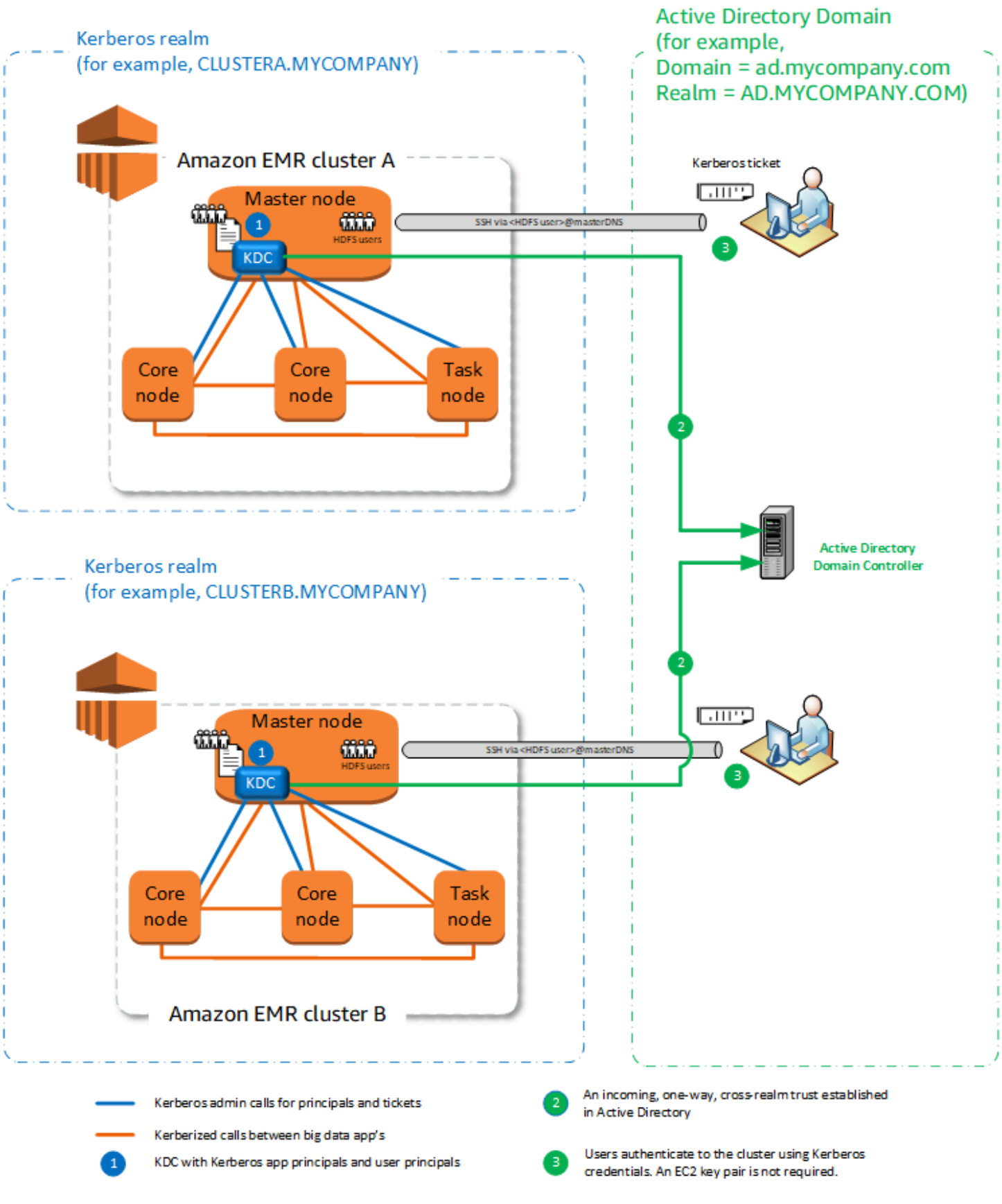
Pertimbangan dan batasan

- Klaster kerberized tidak dapat mengautentikasi satu sama lain, sehingga aplikasi tidak dapat beroperasi. Jika klaster aplikasi perlu untuk beroperasi, Anda harus membuat kepercayaan lintas ranah antara klaster, atau mengatur satu klaster sebagai KDC eksternal untuk klaster lainnya. Jika kepercayaan lintas ranah dibuat, KDC harus memiliki ranah Kerberos yang berbeda.
- Anda harus membuat pengguna Linux pada instance EC2 dari node utama yang sesuai dengan prinsip pengguna KDC, bersama dengan direktori HDFS untuk setiap pengguna.

- Utama harus menggunakan file kunci privat EC2 dan kredensial `kinit` untuk connect ke klaster menggunakan SSH.

Kepercayaan lintas ranah

Di konfigurasi ini, utama (biasanya pengguna) dari ranah Kerberos yang berbeda mengautentikasi komponen aplikasi pada klaster EMR Kerberized, yang memiliki KDC sendiri. KDC pada simpul utama membangun hubungan kepercayaan dengan KDC lain menggunakan prinsip lintas alam yang ada di kedua KDC. Nama utama dan kata sandi cocok di setiap KDC. Kepercayaan lintas ranah yang paling umum dengan implementasi Direktori Aktif, seperti yang ditunjukkan di diagram berikut. Kepercayaan lintas ranah dengan KDC MIT eksternal atau KDC di klaster Amazon EMR lain juga didukung.


Active Directory Domain
(for example,
Domain = ad.mycompany.com
Realm = AD.MYCOMPANY.COM)
Kerberos realm
(for example, CLUSTERA.MYCOMPANY)
Amazon EMR cluster A
Master node
HDFS users
KDC
Core node
Core node
Task node
SSH via <HDFS user>@masterDNS
Kerberos ticket
Active Directory Domain Controller
Kerberos realm
(for example, CLUSTERB.MYCOMPANY)
Amazon EMR cluster B
Kerberos admin calls for principals and tickets
Kerberized calls between big data app's
KDC with Kerberos app principals and user principals
An incoming, one-way, cross-realm trust established in Active Directory
Users authenticate to the cluster using Kerberos credentials. An EC2 key pair is not required.

Keuntungan

- Klaster EMR di mana KDC diinstal mempertahankan kepemilikan penuh KDC.
- Dengan Direktori Aktif, Amazon EMR secara otomatis membuat pengguna Linux yang sesuai dengan utama pengguna dari KDC. Anda masih harus membuat direktori HDFS untuk setiap pengguna. Selain itu, utama pengguna di domain Direktori Aktif dapat mengakses klaster Kerberized menggunakan kredensial `kinit`, tanpa file kunci privat EC2. Ini menghilangkan kebutuhan untuk berbagi file kunci privat di antara pengguna klaster.
- Karena setiap klaster KDC mengelola autentikasi untuk simpul di klaster, efek latensi jaringan dan pengolahan overhead untuk sejumlah besar simpul di klaster diminimalkan.

Pertimbangan dan batasan

- Jika Anda membangun kepercayaan dengan bidang Direktori Aktif, Anda harus memberikan nama pengguna dan kata sandi Direktori Aktif dengan izin untuk menggabungkan utama untuk domain ketika Anda membuat klaster.
- Kepercayaan lintas ranah tidak dapat dibuat antara ranah Kerberos dengan nama yang sama.
- Kepercayaan lintas ranah harus ditetapkan secara eksplisit. Misalnya, jika klaster A dan klaster B keduanya membuat kepercayaan lintas ranah dengan KDC, mereka tidak secara inheren percaya satu sama lain dan aplikasi mereka tidak dapat mengautentikasi satu sama lain untuk beroperasi.
- KDC harus dipelihara secara independen dan terkoordinasi sehingga kredensial utama pengguna cocok.

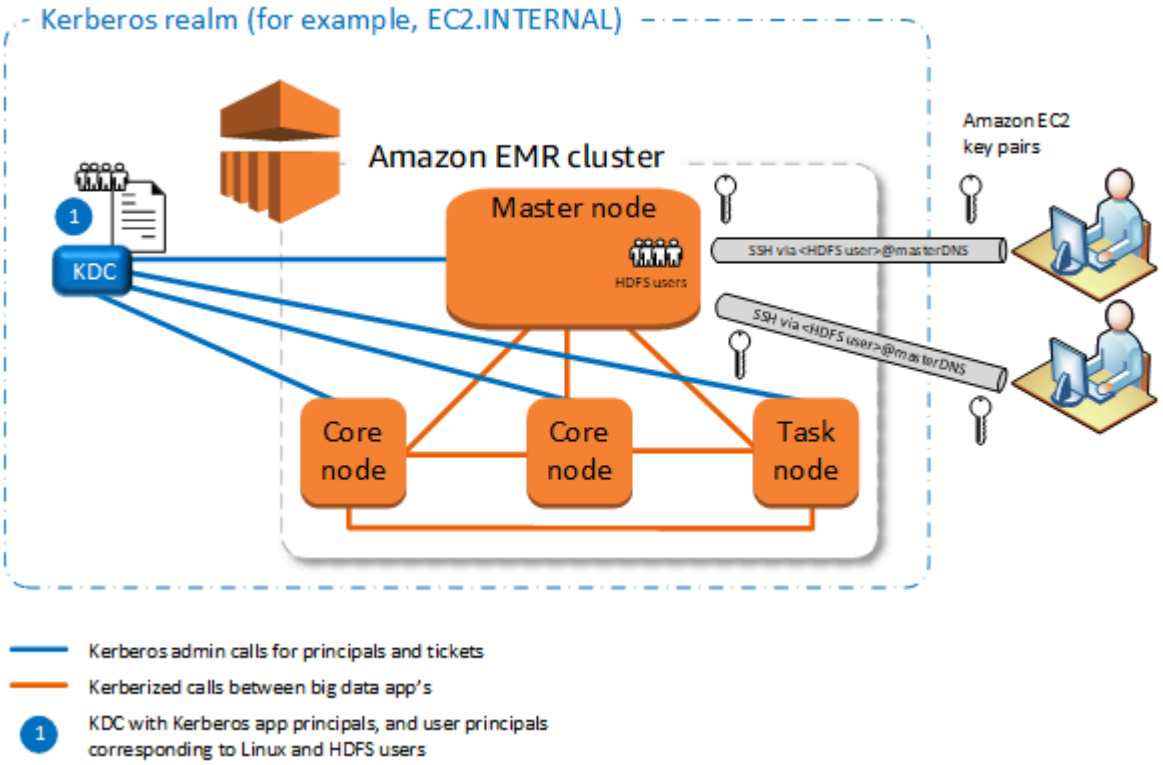
KDC eksternal

Konfigurasi dengan KDC eksternal didukung dengan Amazon EMR 5.20.0 dan versi terbaru.

- KDC Eksternal—KDC MIT
- KDC eksternal — Node primer pada cluster yang berbeda
- KDC eksternal—KDC klaster di klaster yang berbeda dengan kepercayaan lintas ranah Direktori Aktif

KDC Eksternal—KDC MIT

Konfigurasi ini mengizinkan satu klaster EMR atau lebih untuk menggunakan utama didefinisikan dan dipelihara di server KDC MIT.



Keuntungan

- Utama pengelola dikonsolidasikan di satu KDC.
- Beberapa klaster dapat menggunakan KDC yang sama di ranah Kerberos yang sama. Untuk informasi selengkapnya, lihat Persyaratan untuk menggunakan beberapa cluster dengan KDC yang sama.
- Node utama pada cluster Kerberized tidak memiliki beban kinerja yang terkait dengan pemeliharaan KDC.

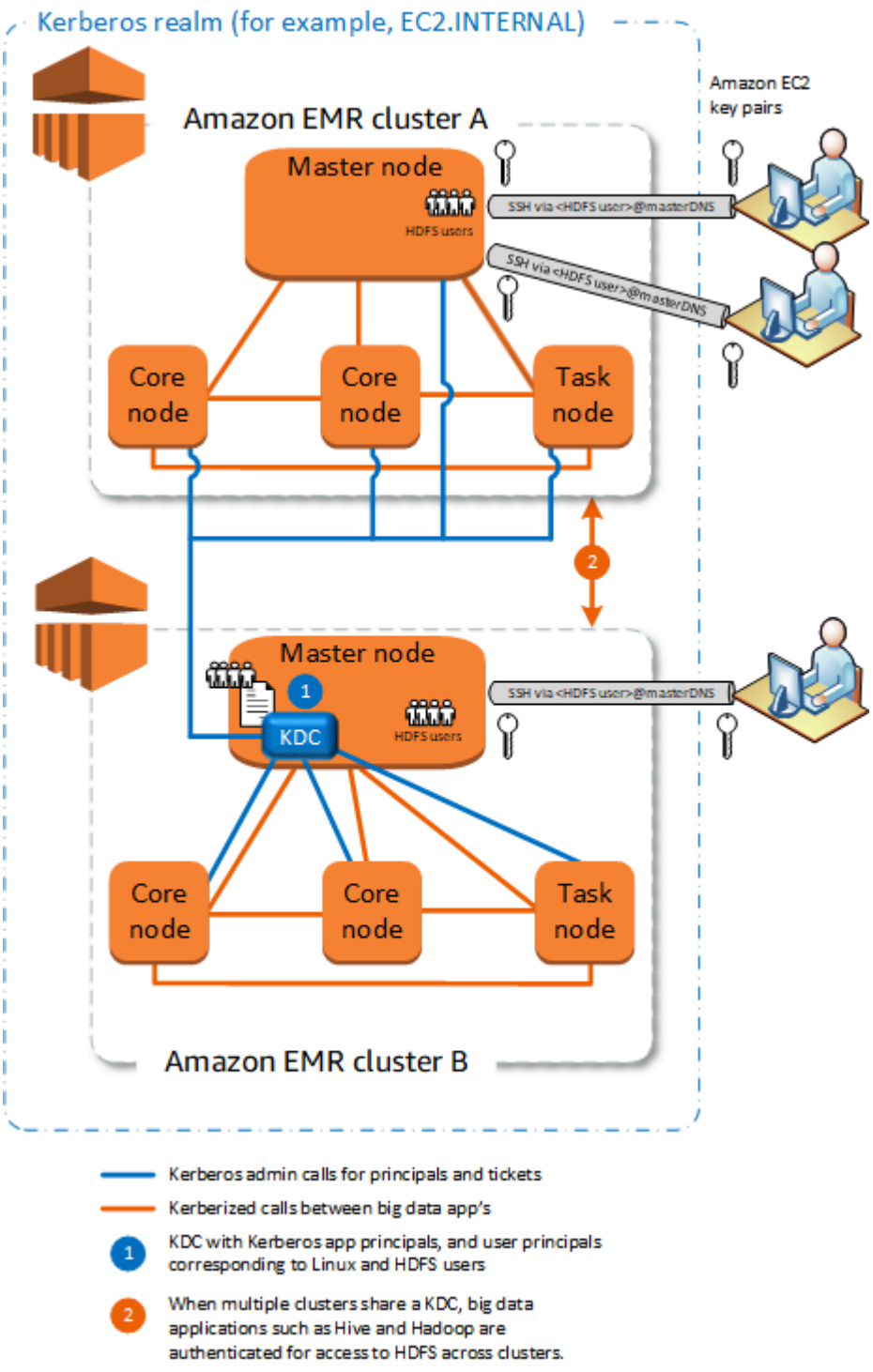
Pertimbangan dan batasan

- Anda harus membuat pengguna Linux pada instance EC2 dari setiap node utama cluster Kerberized yang sesuai dengan prinsip pengguna KDC, bersama dengan direktori HDFS untuk setiap pengguna.
- Utama pengguna harus menggunakan file kunci privat EC2 dan kredensial `kinit` untuk connect ke klaster Kerberized menggunakan SSH.
- Setiap simpul di klaster EMR Kerberized harus memiliki rute jaringan ke KDC.

- Setiap simpul di klaster Kerberized menempatkan beban autentikasi pada KDC eksternal, sehingga konfigurasi KDC mempengaruhi performa klaster. Bila Anda mengonfigurasi perangkat keras server KDC, pertimbangkan jumlah maksimum simpul Amazon EMR yang akan didukung secara bersamaan.
- Klaster performa tergantung pada latensi jaringan antara simpul di klaster Kerberized dan KDC.
- Pemecahan masalah bisa lebih sulit karena saling ketergantungan.

KDC eksternal — Node primer pada cluster yang berbeda

Konfigurasi ini hampir identik dengan implementasi MIT KDC eksternal di atas, kecuali bahwa KDC berada di simpul utama cluster EMR. Untuk informasi selengkapnya, silakan lihat KDC khusus cluster (KDC pada simpul utama) dan Tutorial: Konfigurasi kepercayaan lintas ranah dengan domain Direktori Aktif.



Keuntungan

- Utama pengelola dikonsolidasikan di satu KDC.

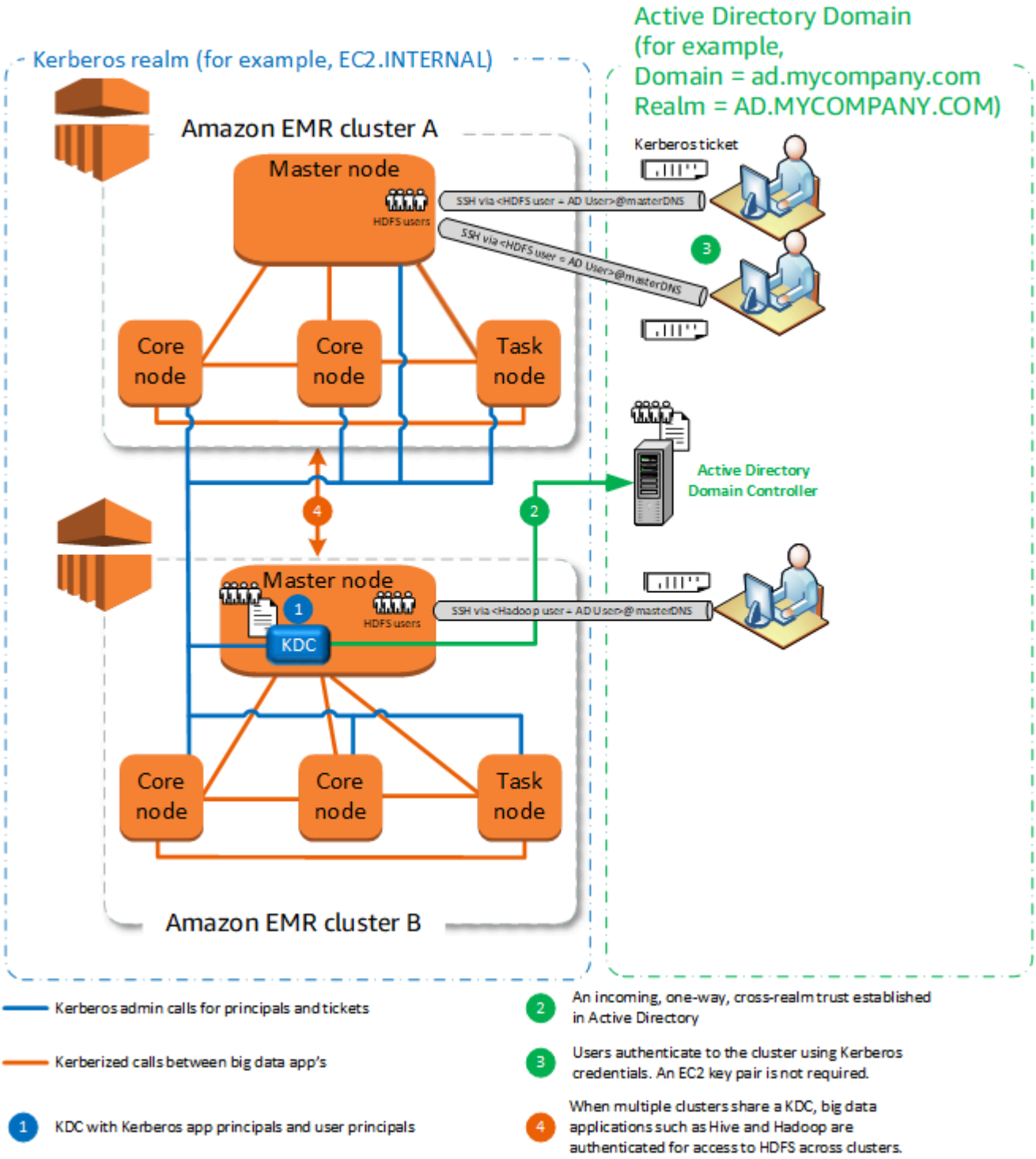
- Beberapa klaster dapat menggunakan KDC yang sama di ranah Kerberos yang sama. Untuk informasi selengkapnya, lihat Persyaratan untuk menggunakan beberapa cluster dengan KDC yang sama.

Pertimbangan dan batasan

- Anda harus membuat pengguna Linux pada instance EC2 dari setiap node utama cluster Kerberized yang sesuai dengan prinsip pengguna KDC, bersama dengan direktori HDFS untuk setiap pengguna.
- Utama pengguna harus menggunakan file kunci privat EC2 dan kredensial `kinit` untuk connect ke klaster Kerberized menggunakan SSH.
- Setiap simpul di setiap klaster EMR harus memiliki rute jaringan ke KDC.
- Setiap simpul Amazon EMR di grup Kerberized menempatkan beban autentikasi pada KDC eksternal, sehingga konfigurasi KDC mempengaruhi performa klaster. Bila Anda mengonfigurasi perangkat keras server KDC, pertimbangkan jumlah maksimum simpul Amazon EMR yang akan didukung secara bersamaan.
- Klaster performa tergantung pada latensi jaringan antara simpul di klaster dan KDC.
- Pemecahan masalah bisa lebih sulit karena saling ketergantungan.

KDC eksternal—KDC klaster di klaster yang berbeda dengan kepercayaan lintas ranah Direktori Aktif

Di konfigurasi ini, Anda pertama kali membuat sebuah klaster dengan KDC khusus klaster yang memiliki satu arah lintas ranah kepercayaan dengan Direktori Aktif. Untuk tutorial detail, lihat Tutorial: Konfigurasi kepercayaan lintas ranah dengan domain Direktori Aktif. Anda kemudian meluncurkan klaster tambahan, referensi KDC klaster yang memiliki kepercayaan sebagai KDC eksternal. Misalnya, lihat KDC klaster eksternal dengan kepercayaan lintas ranah Direktori Aktif. Hal ini mengizinkan setiap klaster Amazon EMR yang menggunakan KDC eksternal untuk mengautentikasi kepala didefinisikan dan dipelihara di domain Direktori Aktif Microsoft.



Keuntungan

- Utama pengelola dikonsolidasikan di domain Direktori Aktif.

- Amazon EMR bergabung dengan ranah Direktori Aktif, yang menghilangkan kebutuhan untuk membuat pengguna Linux yang sesuai dengan pengguna Direktori Aktif. Anda masih harus membuat direktori HDFS untuk setiap pengguna.
- Beberapa klaster dapat menggunakan KDC yang sama di ranah Kerberos yang sama. Untuk informasi selengkapnya, lihat Persyaratan untuk menggunakan beberapa cluster dengan KDC yang sama.
- Utama pengguna di domain Direktori Aktif dapat mengakses klaster Kerberized menggunakan kredensial `kinit`, tanpa file kunci privat EC2. Ini menghilangkan kebutuhan untuk berbagi file kunci privat di antara pengguna klaster.
- Hanya satu simpul utama EMR Amazon yang memiliki beban untuk mempertahankan KDC, dan hanya cluster itu yang harus dibuat dengan kredensyal Direktori Aktif untuk kepercayaan lintas alam antara KDC dan Direktori Aktif.

Pertimbangan dan batasan

- Setiap simpul di setiap klaster EMR harus memiliki rute jaringan ke KDC dan pengendali domain Direktori Aktif.
- Setiap simpul Amazon EMR menempatkan beban autentikasi pada KDC eksternal, sehingga konfigurasi KDC mempengaruhi performa klaster. Bila Anda mengonfigurasi perangkat keras server KDC, pertimbangkan jumlah maksimum simpul Amazon EMR yang akan didukung secara bersamaan.
- Klaster performa tergantung pada latensi jaringan antara simpul di klaster dan server KDC.
- Pemecahan masalah bisa lebih sulit karena saling ketergantungan.

Persyaratan untuk menggunakan beberapa cluster dengan KDC yang sama

Beberapa klaster dapat menggunakan KDC yang sama di ranah Kerberos yang sama. Namun, jika cluster berjalan secara bersamaan, maka cluster mungkin gagal jika mereka menggunakan nama ServicePrincipal Kerberos yang bertentangan.

Jika Anda memiliki beberapa cluster bersamaan dengan KDC eksternal yang sama, maka pastikan bahwa cluster menggunakan alam Kerberos yang berbeda. Jika cluster harus menggunakan ranah Kerberos yang sama, maka pastikan bahwa cluster berada dalam subnet yang berbeda, dan rentang CIDR mereka tidak tumpang tindih.

Mengonfigurasi Kerberos di Amazon EMR

Bagian ini menyediakan detail konfigurasi dan instans untuk menyiapkan Kerberos dengan arsitektur umum. Terlepas dari arsitektur yang Anda pilih, dasar-dasar konfigurasinya sama dan dilakukan di tiga langkah. Jika Anda menggunakan KDC eksternal atau mengatur kepercayaan lintas ranah, Anda harus memastikan bahwa setiap simpul di sebuah klaster memiliki rute jaringan ke KDC eksternal, termasuk konfigurasi grup keamanan yang berlaku untuk mengizinkan lalu lintas Kerberos inbound dan outbound.

Langkah 1: Membuat konfigurasi keamanan dengan properti Kerberos

Konfigurasi keamanan menentukan detail tentang KDC Kerberos, dan mengizinkan konfigurasi Kerberos untuk digunakan kembali setiap kali Anda membuat sebuah klaster. Anda dapat membuat konfigurasi keamanan menggunakan konsol Amazon EMR, AWS CLI, atau API EMR. Konfigurasi keamanan juga dapat berisi opsi keamanan lainnya, seperti enkripsi. Untuk informasi lebih lanjut tentang membuat konfigurasi keamanan dan menentukan konfigurasi keamanan ketika Anda membuat sebuah klaster, lihat Menggunakan konfigurasi keamanan untuk mengatur keamanan klaster. Untuk informasi tentang properti Kerberos di konfigurasi keamanan, lihat Pengaturan Kerberos untuk konfigurasi keamanan.

Langkah 2: Membuat sebuah klaster dan menentukan atribut Kerberos khusus klaster

Ketika Anda membuat sebuah klaster, Anda menentukan konfigurasi keamanan Kerberos bersama dengan pilihan Kerberos khusus klaster. Ketika Anda menggunakan konsol Amazon EMR, hanya opsi Kerberos yang kompatibel dengan konfigurasi keamanan tertentu yang tersedia. Saat Anda menggunakan opsi AWS CLI atau API Amazon EMR, memastikan bahwa Anda menentukan opsi Kerberos kompatibel dengan konfigurasi keamanan yang ditentukan. Misalnya, jika Anda menetapkan kata sandi utama untuk kepercayaan lintas ranah ketika Anda membuat sebuah klaster menggunakan CLI, dan konfigurasi keamanan yang ditentukan tidak dikonfigurasi dengan lintas ranah kepercayaan parameter, maka kesalahan akan terjadi. Untuk informasi selengkapnya, lihat Pengaturan Kerberos untuk klaster.

Langkah 3: Konfigurasikan simpul utama cluster

Tergantung pada persyaratan arsitektur dan implementasi Anda, tambahan set up pada klaster mungkin diperlukan. Anda dapat melakukan ini setelah Anda membuatnya atau menggunakan langkah-langkah atau tindakan bootstrap selama proses pembuatan.

Untuk setiap pengguna yang diautentikasi KerberOS yang terhubung ke cluster menggunakan SSH, Anda harus memastikan bahwa akun Linux dibuat yang sesuai dengan pengguna Kerberos. Jika

prinsipal pengguna disediakan oleh pengontrol domain Active Directory, baik sebagai KDC eksternal atau melalui kepercayaan lintas alam, Amazon EMR membuat akun Linux secara otomatis. Jika Direktori Aktif tidak digunakan, Anda harus membuat utama untuk setiap pengguna yang sesuai dengan pengguna Linux mereka. Untuk informasi selengkapnya, lihat Mengonfigurasi sebuah klaster untuk pengguna HDFS terautentikasi Kerberos dan koneksi SSH.

Setiap pengguna juga harus memiliki direktori pengguna HDFS yang mereka miliki, yang harus Anda buat. Selain itu, SSH harus dikonfigurasi dengan GSSAPI yang diaktifkan untuk mengizinkan koneksi dari pengguna terautentikasi Kerberos. GSSAPI harus diaktifkan pada node utama, dan aplikasi SSH klien harus dikonfigurasi untuk menggunakan GSSAPI. Untuk informasi selengkapnya, lihat Mengonfigurasi sebuah klaster untuk pengguna HDFS terautentikasi Kerberos dan koneksi SSH.

Pengaturan konfigurasi keamanan dan klaster untuk Kerberos di Amazon EMR

Ketika Anda membuat sebuah klaster Kerberized, Anda menentukan konfigurasi keamanan bersama-sama dengan atribut Kerberos yang khusus untuk klaster. Anda tidak dapat menentukan satu set tanpa yang lain, atau akan terjadi kesalahan.

Topik ini menyediakan parameter konfigurasi gambaran umum yang tersedia untuk Kerberos ketika Anda membuat konfigurasi keamanan dan sebuah klaster. Selain itu, contoh CLI untuk membuat konfigurasi keamanan yang kompatibel dan klaster disediakan untuk arsitektur umum.

Pengaturan Kerberos untuk konfigurasi keamanan

Anda dapat membuat konfigurasi keamanan yang menentukan atribut Kerberos menggunakan konsol Amazon EMR, AWS CLI, atau API EMR. Konfigurasi keamanan juga dapat berisi opsi keamanan lainnya, seperti enkripsi. Untuk informasi selengkapnya, lihat Membuat konfigurasi keamanan.

Gunakan referensi berikut untuk memahami pengaturan konfigurasi keamanan yang tersedia untuk arsitektur Kerberos yang Anda pilih. Pengaturan konsol Amazon EMR ditampilkan. Untuk opsi CLI yang sesuai, lihat Menentukan pengaturan Kerberos menggunakan AWS CLI atau Contoh konfigurasi.

Parameter	Deskripsi
Kerberos	Menentukan bahwa Kerberos diaktifkan untuk klaster yang menggunakan konfigurasi keamanan ini. Jika sebuah klaster menggunakan konfigurasi keamanan

Parameter		Deskripsi
		ini, klaster juga harus memiliki pengaturan Kerberos yang ditentukan atau terjadi kesalahan.
Penyedia	KDC khusus cluster	Menentukan bahwa Amazon EMR membuat KDC pada node utama dari setiap cluster yang menggunak an konfigurasi keamanan ini. Anda menentukan nama ranah dan kata sandi admin KDC ketika Anda membuat klaster. Anda dapat referensi KDC ini dari klaster lain, jika diperlukan. Membuat klaster tersebut menggunakan konfigurasi keamanan yang berbeda, menentukan KDC eksternal, dan menggunakan nama ranah dan kata sandi admin KDC yang Anda tentukan untuk KDC khusus klaster.
	KDC Eksternal	Hanya tersedia dengan Amazon EMR 5.20.0 dan yang lebih baru. Menentukan bahwa klaster menggunak an konfigurasi keamanan ini mengautentikasi utama Kerberos menggunakan server KDC di luar klaster. KDC tidak dibuat pada klaster. Ketika Anda membuat klaster, Anda menentukan nama ranah dan kata sandi admin KDC untuk KDC eksternal.
Tiket Seumur Hidup		Opsional. Menentukan periode tiket Kerberos mana yang valid yang dikeluarkan oleh KDC pada klaster yang menggunakan konfigurasi keamanan ini. Masa pakai tiket terbatas untuk alasan keamanan. Aplikasi klaster dan layanan perpanjangan tiket otomatis setelah mereka kedaluwarsa. Pengguna yang terhubung ke cluster melalui SSH menggunakan kredensyal Kerberos harus menjalankan `kinit` dari baris perintah node utama untuk memperbarui setelah tiket kedaluwarsa.

Parameter		Deskripsi
Kepercayaan lintas alam		Menentukan kepercayaan lintas ranah antara KDC khusus klaster pada klaster yang menggunakan konfigurasi keamanan ini dan KDC di ranah Kerberos yang berbeda. Utama (biasanya pengguna) dari ranah lain diautenti kasi ke klaster yang menggunakan konfigurasi ini. Konfigurasi tambahan di ranah Kerberos lainnya diperlukan. Untuk informasi selengkapnya, lihat Tutorial: Konfigurasi kepercayaan lintas ranah dengan domain Direktori Aktif.
Properti kepercaya an lintas ranah	Realm	Menentukan nama ranah Kerberos dari ranah lain di hubungan kepercayaan. Dengan konvensi, nama ranah Kerberos adalah sama dengan nama domain tetapi semuanya menggunakan huruf kapital.
	Domain	Menentukan nama domain dari ranah lain di hubungan kepercayaan.
	Server admin	Menentukan nama domain yang memenuhi syarat (FQDN) atau alamat IP dari server admin di ranah lain dari hubungan kepercayaan. server admin dan server KDC biasanya berjalan pada mesin yang sama dengan FQDN yang sama, tetapi berkomunikasi pada port yang berbeda. Jika port tidak ditentukan, port 749 digunakan, yang merupakan default Kerberos. Atau, Anda dapat menentukan port (misalnya, `domain.example.com:749`).

Parameter		Deskripsi
	Server KDC	Menentukan nama domain yang memenuhi syarat (FQDN) atau alamat IP server KDC di ranah lain dari hubungan kepercayaan. Server KDC dan server admin biasanya berjalan pada mesin yang sama dengan FQDN yang sama, tetapi menggunakan port yang berbeda. Jika port tidak ditentukan, port 88 digunakan, yang merupakan default Kerberos. Atau, Anda dapat menentukan port (misalnya, `domain.example.com:88`).
KDC Eksternal		Menentukan bahwa klaster eksternal KDC digunakan oleh klaster.
Properti KDC eksternal	Server admin	Menentukan nama domain yang memenuhi syarat (FQDN) atau alamat IP dari server admin eksternal. Server admin dan server KDC biasanya berjalan pada mesin yang sama dengan FQDN yang sama, tetapi berkomunikasi pada port yang berbeda. Jika port tidak ditentukan, port 749 digunakan, yang merupakan default Kerberos. Atau, Anda dapat menentukan port (misalnya, `domain.example.com:749`).
	Server KDC	Menentukan nama domain yang memenuhi syarat (FQDN) dari server KDC eksternal. Server KDC dan server admin biasanya berjalan pada mesin yang sama dengan FQDN yang sama, tetapi menggunakan port yang berbeda. Jika port tidak ditentukan, port 88 digunakan, yang merupakan default Kerberos. Atau, Anda dapat menentukan port (misalnya, `domain.example.com:88`).

Parameter			Deskripsi
	Integrasi Direktori Aktif		Menentukan bahwa autentikasi utama Kerberos terintegrasi dengan domain Direktori Aktif Microsoft.
	Properti integrasi Direktori Aktif	Ranah Direktori Aktif	Menentukan nama ranah Kerberos dari domain Direktori Aktif. Dengan konvensi, nama ranah Kerberos biasanya sama dengan nama domain tetapi di huruf kapital semua.
		Domain Direktori Aktif	Menentukan nama domain Direktori Aktif.
		Server Direktori Aktif	Menentukan nama domain yang memenuhi syarat (FQDN) dari pengendali domain Direktori Aktif Microsoft.

Pengaturan Kerberos untuk klaster

Anda dapat menentukan pengaturan Kerberos ketika Anda membuat sebuah klaster menggunakan konsol Amazon EMR, AWS CLI, atau API EMR.

Gunakan referensi berikut untuk memahami pengaturan konfigurasi klaster yang tersedia untuk arsitektur Kerberos yang Anda pilih. Pengaturan konsol Amazon EMR ditampilkan. Untuk opsi CLI yang sesuai, lihat Contoh konfigurasi.

Parameter	Deskripsi
Ranah	Nama ranah Kerberos untuk klaster. Konvensi Kerberos adalah untuk mengatur ini agar sama dengan nama domain, tetapi dengan huruf besar. Misalnya, untuk domain `ec2.internal` , menggunakan `EC2.INTERNAL` sebagai nama ranah.
Kata sandi admin KDC	

Parameter	Deskripsi
	Kata sandi yang digunakan di klaster untuk `kadmin` atau `kadmin.local` . Ini adalah antarmuka baris perintah untuk sistem administrasi Kerberos V5, yang mempertah ankan Kerberos utama, kebijakan kata sandi, dan keytabs untuk klaster.
Kepercayaan lintas ranah kata sandi utama (opsional)	Diperlukan saat membangun kepercaya an lintas ranah. Kata sandi utama lintas ranah, yang harus identik di seluruh ranah. Menggunakan kata sandi yang kuat.
Pengguna gabungan domain Direktori Aktif (opsional)	Diperlukan saat menggunakan Direktori Aktif di kepercayaan lintas ranah. Ini adalah nama log in pengguna akun Direktori Aktif dengan izin untuk bergabung dengan komputer ke domain. Amazon EMR menggunakan identitas ini untuk bergabung dengan klaster ke domain. Untuk informasi selengkapnya, lihat the section called “Langkah 3: Tambahkan akun ke domain untuk EMR Cluster”.
Kata sandi gabungan domain Direktori Aktif (opsional)	Kata sandi untuk pengguna gabungan domain Direktori Aktif Untuk informasi selengkapnya, lihat the section called “Langkah 3: Tambahkan akun ke domain untuk EMR Cluster”.

Contoh konfigurasi

Contoh berikut menunjukkan konfigurasi keamanan dan konfigurasi klaster untuk skenario umum. perintah AWS CLI ditampilkan untuk singkatnya.

KDC Lokal

Perintah berikut membuat cluster dengan KDC khusus cluster yang berjalan pada node utama. Konfigurasi tambahan pada klaster diperlukan. Untuk informasi selengkapnya, lihat Mengonfigurasi sebuah klaster untuk pengguna HDFS terautentikasi Kerberos dan koneksi SSH.

Buat Konfigurasi Keamanan

```
aws emr create-security-configuration --name LocalKDCSecurityConfig \
--security-configuration '{"AuthenticationConfiguration": \
{"KerberosConfiguration": {"Provider": "ClusterDedicatedKdc",\
"ClusterDedicatedKdcConfiguration": {"TicketLifetimeInHours": 24 }}}}'
```

Buat Cluster

```
aws emr create-cluster --release-label emr-7.0.0 \
--instance-count 3 --instance-type m5.xlarge \
--applications Name=Hadoop Name=Hive --ec2-attributes
 InstanceProfile=EMR_EC2_DefaultRole,KeyName=MyEC2Key \
--service-role EMR_DefaultRole \
--security-configuration LocalKDCSecurityConfig \
--kerberos-attributes Realm=EC2.INTERNAL,KdcAdminPassword=MyPassword
```

KDC khusus klaster dengan kepercayaan lintas ranah Direktori Aktif

Perintah berikut membuat cluster dengan KDC khusus cluster yang berjalan pada node utama dengan kepercayaan lintas-ranah ke domain Active Directory. Konfigurasi tambahan pada klaster dan Direktori Aktif diperlukan. Untuk informasi selengkapnya, lihat Tutorial: Konfigurasi kepercayaan lintas ranah dengan domain Direktori Aktif.

Buat Konfigurasi Keamanan

```
aws emr create-security-configuration --name LocalKDCWithADTrustSecurityConfig \
--security-configuration '{"AuthenticationConfiguration": \
{"KerberosConfiguration": {"Provider": "ClusterDedicatedKdc", \
"ClusterDedicatedKdcConfiguration": {"TicketLifetimeInHours": 24, \
"CrossRealmTrustConfiguration": {"Realm":"AD.DOMAIN.COM", \
"Domain":"ad.domain.com", "AdminServer":"ad.domain.com", \
"KdcServer":"ad.domain.com"}}}}}'
```

Buat Cluster

```
aws emr create-cluster --release-label emr-7.0.0 \
--instance-count 3 --instance-type m5.xlarge --applications Name=Hadoop Name=Hive \
--ec2-attributes InstanceProfile=EMR_EC2_DefaultRole,KeyName=MyEC2Key \
--service-role EMR_DefaultRole --security-configuration KDCWithADTrustSecurityConfig \
--kerberos-attributes Realm=EC2.INTERNAL,KdcAdminPassword=MyClusterKDCAdminPassword,\
ADDomainJoinUser=ADUserLogonName,ADDomainJoinPassword=ADUserPassword,\
CrossRealmTrustPrincipalPassword=MatchADTrustPassword
```

KDC eksternal pada klaster yang berbeda

Perintah berikut membuat cluster yang mereferensikan KDC khusus cluster pada node utama dari cluster yang berbeda untuk mengautentikasi prinsipal. Konfigurasi tambahan pada klaster diperlukan. Untuk informasi selengkapnya, lihat Mengonfigurasi sebuah klaster untuk pengguna HDFS terautentikasi Kerberos dan koneksi SSH.

Buat Konfigurasi Keamanan

```
aws emr create-security-configuration --name ExtKDCOnDifferentCluster \
--security-configuration '{"AuthenticationConfiguration": \
{"KerberosConfiguration": {"Provider": "ExternalKdc", \
"ExternalKdcConfiguration": {"KdcServerType": "Single", \
"AdminServer": "MasterDNSOfKDCMaster:749", \
"KdcServer": "MasterDNSOfKDCMaster:88"}}}}'
```

Buat Cluster

```
aws emr create-cluster --release-label emr-7.0.0 \
--instance-count 3 --instance-type m5.xlarge \
--applications Name=Hadoop Name=Hive \
--ec2-attributes InstanceProfile=EMR_EC2_DefaultRole,KeyName=MyEC2Key \
--service-role EMR_DefaultRole --security-configuration ExtKDCOnDifferentCluster \
--kerberos-attributes Realm=EC2.INTERNAL,KdcAdminPassword=KDCOnMasterPassword
```

KDC klaster eksternal dengan kepercayaan lintas ranah Direktori Aktif

Perintah berikut membuat klaster tanpa KDC. Cluster mereferensikan KDC khusus cluster yang berjalan pada node utama cluster lain untuk mengautentikasi prinsipal. KDC yang memiliki kepercayaan lintas ranah dengan pengendali domain Direktori Aktif. Konfigurasi tambahan pada node utama dengan KDC diperlukan. Untuk informasi selengkapnya, lihat Tutorial: Konfigurasi kepercayaan lintas ranah dengan domain Direktori Aktif.

Buat Konfigurasi Keamanan

```
aws emr create-security-configuration --name ExtKDCWithADIntegration \
--security-configuration '{"AuthenticationConfiguration": \
{"KerberosConfiguration": {"Provider": "ExternalKdc", \
"ExternalKdcConfiguration": {"KdcServerType": "Single", \
"AdminServer": "MasterDNSofClusterKDC:749", \
"KdcServer": "MasterDNSofClusterKDC.com:88", \
"AdIntegrationConfiguration": {"AdRealm":"AD.DOMAIN.COM", \
"AdDomain":"ad.domain.com", \
"AdServer":"ad.domain.com"}}}}}'
```

Buat Cluster

```
aws emr create-cluster --release-label emr-7.0.0 \
--instance-count 3 --instance-type m5.xlarge --applications Name=Hadoop Name=Hive \
--ec2-attributes InstanceProfile=EMR_EC2_DefaultRole,KeyName=MyEC2Key \
--service-role EMR_DefaultRole --security-configuration ExtKDCWithADIntegration \
--kerberos-attributes Realm=EC2.INTERNAL,KdcAdminPassword=KDCOnMasterPassword,\
ADDomainJoinUser=MyPrivilegedADUserName,ADDomainJoinPassword=PasswordForADDomainJoinUser
```

Mengonfigurasi sebuah klaster untuk pengguna HDFS terautentikasi Kerberos dan koneksi SSH

Amazon EMR membuat klien pengguna terautentikasi Kerberos untuk aplikasi yang berjalan di klaster misalnya, pengguna `hadoop`, pengguna `spark`, dan lainnya. Anda juga dapat menambahkan pengguna yang terautentikasi agar klaster memproses menggunakan Kerberos. Pengguna terautentikasi kemudian dapat connect ke klaster dengan kredensial Kerberos mereka dan bekerja dengan aplikasi. Bagi pengguna yang ingin mengautentikasi ke klaster, konfigurasi berikut diperlukan:

- Akun Linux yang cocok dengan prinsipal Kerberos di KDC harus ada di cluster. Amazon EMR melakukan ini secara otomatis di arsitektur yang mengintegrasikan dengan Direktori Aktif.
- Anda harus membuat direktori pengguna HDFS pada node utama untuk setiap pengguna, dan memberikan izin pengguna ke direktori.
- Anda harus mengkonfigurasi layanan SSH sehingga GSSAPI diaktifkan pada node utama. Selain itu, pengguna harus memiliki klien SSH dengan GSSAPI diaktifkan.

Menambahkan pengguna Linux dan prinsipal Kerberos ke node utama

Jika Anda tidak menggunakan Active Directory, Anda harus membuat akun Linux pada node utama cluster dan menambahkan prinsipal untuk pengguna Linux ini ke KDC. Ini termasuk prinsipal di KDC untuk simpul utama. Selain prinsipal pengguna, KDC yang berjalan pada node primer membutuhkan prinsipal untuk host lokal.

Ketika arsitektur Anda termasuk integrasi Direktori Aktif, pengguna Linux dan utama di KDC lokal, jika berlaku, dibuat secara otomatis. Anda bisa melewati langkah ini. Untuk informasi lebih lanjut, lihat Kepercayaan lintas ranah dan KDC eksternal—KDC klaster di klaster yang berbeda dengan kepercayaan lintas ranah Direktori Aktif.

> **Important**
>
> KDC, bersama dengan database prinsipal, hilang ketika node utama berakhir karena node utama menggunakan penyimpanan sementara. Jika Anda membuat pengguna untuk koneksi SSH, kami merekomendasikan Anda membuat kepercayaan lintas ranah dengan KDC eksternal yang dikonfigurasi untuk ketersediaan tinggi. Atau, jika Anda membuat pengguna untuk koneksi SSH menggunakan akun Linux, otomatiskan proses pembuatan akun menggunakan tindakan dan skrip bootstrap sehingga dapat diulang saat Anda membuat cluster baru.

Mengirimkan langkah ke klaster setelah Anda membuatnya atau ketika Anda membuat klaster adalah cara termudah untuk menambahkan pengguna dan utama KDC. Atau, Anda dapat terhubung ke node utama menggunakan EC2 key pair sebagai `hadoop` pengguna default untuk menjalankan perintah. Untuk informasi selengkapnya, lihat Connect ke node utama menggunakan SSH.

Contoh berikut mengirimkan script bash `configureCluster.sh` untuk sebuah klaster yang sudah ada, mereferensikan ID klaster. Script disimpan ke Amazon S3.

```
aws emr add-steps --cluster-id <j-2AL4XXXXXX5T9> \
--steps Type=CUSTOM_JAR,Name=CustomJAR,ActionOnFailure=CONTINUE,\
Jar=s3://region.elasticmapreduce/libs/script-runner/script-runner.jar,\
Args=["s3://DOC-EXAMPLE-BUCKET/configureCluster.sh"]
```

Contoh berikut menunjukkan isi dari script `configureCluster.sh`. Script juga menangani membuat direktori pengguna HDFS dan mengaktifkan GSSAPI untuk SSH, yang dibahas di bagian berikut.

```
#!/bin/bash
#Add a principal to the KDC for the primary node, using the primary node's returned
 host name
sudo kadmin.local -q "ktadd -k /etc/krb5.keytab host/`hostname -f`"
#Declare an associative array of user names and passwords to add
declare -A arr
arr=([lijuan]=pwd1 [marymajor]=pwd2 [richardroe]=pwd3)
for i in ${!arr[@]}; do
    #Assign plain language variables for clarity
     name=${i}
     password=${arr[${i}]}

     # Create a principal for each user in the primary node and require a new password
 on first logon
     sudo kadmin.local -q "addprinc -pw $password +needchange $name"

     #Add hdfs directory for each user
     hdfs dfs -mkdir /user/$name

     #Change owner of each user's hdfs directory to that user
     hdfs dfs -chown $name:$name /user/$name
done

# Enable GSSAPI authentication for SSH and restart SSH service
sudo sed -i 's/^.*GSSAPIAuthentication.*$/GSSAPIAuthentication yes/' /etc/ssh/
sshd_config
sudo sed -i 's/^.*GSSAPICleanupCredentials.*$/GSSAPICleanupCredentials yes/' /etc/ssh/
sshd_config
sudo systemctl restart sshd
```

Menambahkan direktori HDFS pengguna

Untuk memungkinkan pengguna Anda masuk ke cluster untuk menjalankan pekerjaan Hadoop, Anda harus menambahkan direktori pengguna HDFS untuk akun Linux mereka, dan memberikan setiap pengguna kepemilikan direktori mereka.

Mengirimkan langkah ke klaster setelah Anda membuat atau ketika Anda membuat klaster adalah cara termudah untuk membuat direktori HDFS. Atau, Anda dapat terhubung ke node utama menggunakan EC2 key pair sebagai `hadoop` pengguna default untuk menjalankan perintah. Untuk informasi selengkapnya, lihat Connect ke node utama menggunakan SSH.

Contoh berikut mengirimkan script bash `AddHDFSUsers.sh` untuk sebuah klaster yang sudah ada, mereferensikan ID klaster. Script disimpan ke Amazon S3.

```
aws emr add-steps --cluster-id <j-2AL4XXXXXX5T9> \
--steps Type=CUSTOM_JAR,Name=CustomJAR,ActionOnFailure=CONTINUE,\
Jar=s3://region.elasticmapreduce/libs/script-runner/script-runner.jar,Args=["s3://DOC-EXAMPLE-BUCKET/AddHDFSUsers.sh"]
```

Contoh berikut menunjukkan isi dari script `AddHDFSUsers.sh`.

```
#!/bin/bash
# AddHDFSUsers.sh script

# Initialize an array of user names from AD, or Linux users created manually on the
 cluster
ADUSERS=("lijuan" "marymajor" "richardroe" "myusername")

# For each user listed, create an HDFS user directory
# and change ownership to the user

for username in ${ADUSERS[@]}; do
     hdfs dfs -mkdir /user/$username
     hdfs dfs -chown $username:$username /user/$username
done
```

Mengaktifkan GSSAPI untuk SSH

Agar pengguna yang diautentikasi KerberOS dapat terhubung ke node utama menggunakan SSH, layanan SSH harus mengaktifkan otentikasi GSSAPI. Untuk mengaktifkan GSSAPI, jalankan perintah berikut dari baris perintah node utama atau gunakan langkah untuk menjalankannya sebagai skrip. Setelah mengonfigurasi ulang SSH, Anda harus me-restart layanan.

```
sudo sed -i 's/^.*GSSAPIAuthentication.*$/GSSAPIAuthentication yes/' /etc/ssh/
sshd_config
sudo sed -i 's/^.*GSSAPICleanupCredentials.*$/GSSAPICleanupCredentials yes/' /etc/ssh/
sshd_config
sudo systemctl restart sshd
```

Menggunakan SSH untuk connect ke klaster Kerberized

Bagian ini menunjukkan langkah-langkah bagi pengguna yang diautentikasi KerberOS untuk terhubung ke simpul utama cluster EMR.

Setiap komputer yang digunakan untuk koneksi SSH harus menginstal klien SSH dan aplikasi klien Kerberos. Komputer Linux kemungkinan besar ikut memasukkan ini secara default. Misalnya, OpenSSH diinstal pada kebanyakan sistem operasi Linux, Unix, dan macOS. Anda dapat memeriksa klien SSH dengan mengetik ssh di baris perintah. Jika komputer Anda tidak mengenali perintah, instal klien SSH untuk terhubung ke node utama. Proyek OpenSSH menyediakan implementasi gratis rangkaian lengkap alat SSH. Untuk informasi selengkapnya, lihat situs web OpenSSH. Pengguna Windows dapat menggunakan aplikasi seperti PuTTY T sebagai klien SSH.

Untuk informasi selengkapnya tentang koneksi SSH, lihat Connect ke sebuah cluster.

SSH menggunakan GSSAPI untuk mengautentikasi klien Kerberos, dan Anda harus mengaktifkan otentikasi GSSAPI untuk layanan SSH pada node utama cluster. Untuk informasi selengkapnya, lihat Mengaktifkan GSSAPI untuk SSH. Klien SSH juga harus menggunakan GSSAPI.

Dalam contoh berikut, untuk MasterPublicDNS gunakan nilai yang muncul untuk ***Master public DNS*** *pada tab* ***Ringkasan*** *panel detail klaster—misalnya, ec2-11-222-33-44.compute-1.amazonaws.com.*

Prasyarat untuk krb5.conf (Bukan Direktori Aktif)

Saat menggunakan konfigurasi tanpa integrasi Active Directory, selain klien SSH dan aplikasi klien Kerberos, setiap komputer klien harus memiliki salinan `/etc/krb5.conf` file yang cocok dengan `/etc/krb5.conf` file pada node primer cluster.

Untuk menyalin file krb5.conf

1. Gunakan SSH untuk terhubung ke node utama menggunakan key pair EC2 dan `hadoop` pengguna default—misalnya,. `hadoop@`*`MasterPublicDNS`* Untuk instruksi detail, lihat Connect ke sebuah cluster.
2. Dari simpul utama, salin isi `/etc/krb5.conf` file. Untuk informasi selengkapnya, lihat Connect ke sebuah cluster.
3. Pada setiap komputer klien yang akan connect ke klaster, buat file `/etc/krb5.conf` identik berdasarkan salinan yang Anda buat pada langkah sebelumnya.

Menggunakan kinit dan SSH

Setiap kali pengguna connect dari komputer klien menggunakan kredensial Kerberos, pengguna harus terlebih dahulu memperbaharui tiket Kerberos untuk pengguna mereka pada komputer klien. Selain itu, klien SSH harus dikonfigurasi untuk menggunakan autentikasi GSSAPI.

Untuk menggunakan SSH agar connect ke klaster EMR Kerberized

1. Gunakan `kinit` untuk memperbarui tiket Kerberos seperti yang ditunjukkan di contoh berikut

   ```
   kinit user1
   ```

2. Gunakan klien `ssh` bersama dengan utama yang Anda buat di KDC khusus klaster atau nama pengguna Direktori Aktif. Pastikan bahwa autentikasi GSSAPI diaktifkan seperti yang ditunjukkan di contoh berikut.

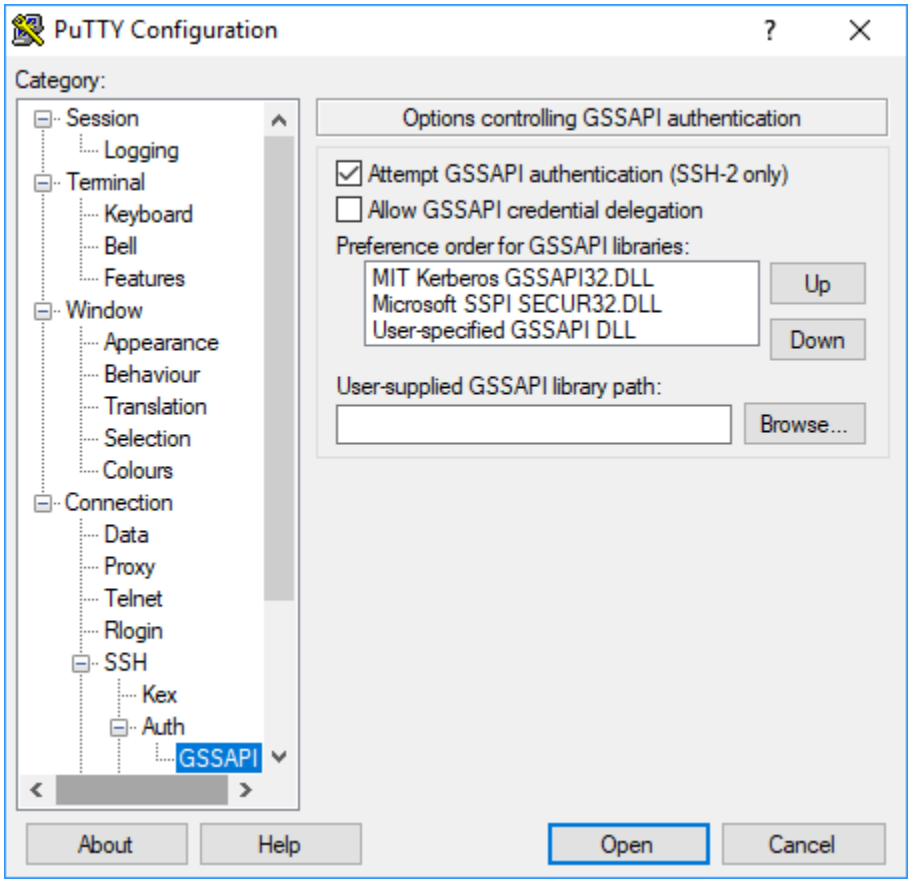
   Contoh: Pengguna Linux

   Opsi `-K` menentukan autentikasi GSSAPI.

   ```
   ssh -K user1@MasterPublicDNS
   ```

   Contoh: Pengguna Windows (PutTY)

   Pastikan bahwa opsi autentikasi GSSAPI untuk sesi diaktifkan seperti yang ditunjukkan:



Tutorial: mengonfigurasi KDC khusus klaster

Topik ini memandu Anda melalui pembuatan cluster dengan pusat distribusi kunci khusus cluster (KDC), menambahkan akun Linux secara manual ke semua node cluster, menambahkan prinsip Kerberos ke KDC pada node utama, dan memastikan bahwa komputer klien memiliki klien Kerberos diinstal.

Untuk informasi lebih lanjut tentang support Amazon EMR untuk Kerberos dan KDC, serta tautan ke Dokumentasi MIT Kerberos, lihat Gunakan Kerberos untuk otentikasi dengan Amazon EMR.

Langkah 1: Buat klaster Kerberized

1. Buat konfigurasi keamanan yang mengaktifkan Kerberos. Contoh berikut menunjukkan perintah `create-security-configuration` menggunakan AWS CLI yang menentukan konfigurasi keamanan sebagai struktur JSON inline. Anda juga dapat membuat referensi pada file yang disimpan secara lokal.

```
aws emr create-security-configuration --name MyKerberosConfig \
```

```
--security-configuration '{"AuthenticationConfiguration": {"KerberosConfiguration":
{"Provider": "ClusterDedicatedKdc", "ClusterDedicatedKdcConfiguration":
 {"TicketLifetimeInHours": 24}}}}'
```

2. Buat sebuah klaster yang membuat referensi pada konfigurasi keamanan, menetapkan atribut Kerberos untuk klaster, dan menambahkan akun Linux menggunakan tindakan bootstrap. Contoh berikut menunjukkan perintah `create-cluster` menggunakan AWS CLI. Perintah referensi konfigurasi keamanan yang Anda buat di atas, `MyKerberosConfig`. Itu juga membuat referensi script sederhana, `createlinuxusers.sh`, sebagai tindakan bootstrap, yang Anda buat dan unggah ke Amazon S3 sebelum membuat klaster.

```
aws emr create-cluster --name "MyKerberosCluster" \
--release-label emr-7.0.0 \
--instance-type m5.xlarge \
--instance-count 3 \
--ec2-attributes InstanceProfile=EMR_EC2_DefaultRole,KeyName=MyEC2KeyPair \
--service-role EMR_DefaultRole \
--security-configuration MyKerberosConfig \
--applications Name=Hadoop Name=Hive Name=Oozie Name=Hue Name=HCatalog Name=Spark \
--kerberos-attributes Realm=EC2.INTERNAL,\
KdcAdminPassword=MyClusterKDCAdminPwd \
--bootstrap-actions Path=s3://DOC-EXAMPLE-BUCKET/createlinuxusers.sh
```

   Kode berikut menunjukkan isi `createlinuxusers.sh` skrip, yang menambahkan user1, user2, dan user3 ke setiap node di cluster. Pada langkah berikutnya, Anda menambahkan pengguna ini sebagai utama KDC.

```
#!/bin/bash
sudo adduser user1
sudo adduser user2
sudo adduser user3
```

Langkah 2: Menambahkan utama ke KDC, membuat direktori pengguna HDFS, dan mengonfigurasi SSH

KDC yang berjalan pada node primer membutuhkan prinsipal yang ditambahkan untuk host lokal dan untuk setiap pengguna yang Anda buat di cluster. Anda juga dapat membuat direktori HDFS untuk setiap pengguna jika mereka perlu untuk connect ke klaster dan menjalankan Tugas Hadoop.

Demikian pula, konfigurasi layanan SSH untuk mengaktifkan autentikasi GSSAPI, yang diperlukan untuk Kerberos. Setelah Anda mengaktifkan GSSAPI, restart layanan SSH.

Cara termudah untuk menyelesaikan tugas-tugas ini adalah kirim langkah ke klaster. Contoh berikut kirim ke script bash `configurekdc.sh` untuk klaster yang Anda buat pada langkah sebelumnya, mereferensikan ID klasternya. Script disimpan ke Amazon S3. Atau, Anda dapat terhubung ke node utama menggunakan key pair EC2 untuk menjalankan perintah atau mengirimkan langkah selama pembuatan cluster.

```
aws emr add-steps --cluster-id <j-2AL4XXXXXX5T9> --steps
 Type=CUSTOM_JAR,Name=CustomJAR,ActionOnFailure=CONTINUE,Jar=s3://
myregion.elasticmapreduce/libs/script-runner/script-runner.jar,Args=["s3://DOC-EXAMPLE-
BUCKET/configurekdc.sh"]
```

Kode berikut menunjukkan isi `configurekdc.sh` skrip.

```
#!/bin/bash
#Add a principal to the KDC for the primary node, using the primary node's returned
 host name
sudo kadmin.local -q "ktadd -k /etc/krb5.keytab host/`hostname -f`"
#Declare an associative array of user names and passwords to add
declare -A arr
arr=([user1]=pwd1 [user2]=pwd2 [user3]=pwd3)
for i in ${!arr[@]}; do
    #Assign plain language variables for clarity
     name=${i}
     password=${arr[${i}]}

     # Create principal for sshuser in the primary node and require a new password on
 first logon
     sudo kadmin.local -q "addprinc -pw $password +needchange $name"

     #Add user hdfs directory
     hdfs dfs -mkdir /user/$name

     #Change owner of user's hdfs directory to user
     hdfs dfs -chown $name:$name /user/$name
done

# Enable GSSAPI authentication for SSH and restart SSH service
sudo sed -i 's/^.*GSSAPIAuthentication.*$/GSSAPIAuthentication yes/' /etc/ssh/
sshd_config
```

```
sudo sed -i 's/^.*GSSAPICleanupCredentials.*$/GSSAPICleanupCredentials yes/' /etc/ssh/
sshd_config
sudo systemctl restart sshd
```

Pengguna yang Anda tambahkan sekarang dapat connect ke klaster menggunakan SSH. Untuk informasi selengkapnya, lihat Menggunakan SSH untuk connect ke klaster Kerberized.

Tutorial: Konfigurasi kepercayaan lintas ranah dengan domain Direktori Aktif

Ketika Anda mengatur kepercayaan lintas ranah, Anda mengizinkan utama (biasanya pengguna) dari ranah Kerberos yang berbeda untuk mengautentikasi komponen aplikasi pada klaster EMR. Pusat distribusi kunci (KDC) khusus klaster menetapkan hubungan kepercayaan dengan KDC lain menggunakan utama lintas ranah yang ada di kedua KDC. Nama utama dan kata sandi sangat cocok.

Kepercayaan lintas ranah mengharuskan KDC dapat mencapai satu sama lain melalui jaringan dan menyelesaikan nama domain masing-masing. Langkah-langkah untuk membangun hubungan kepercayaan lintas ranah dengan pengendali domain Microsoft AD berjalan sebagai instans EC2 disediakan di bawah ini, bersama dengan pengaturan jaringan contoh yang menyediakan konektivitas dan resolusi nama domain yang diperlukan. Setiap pengaturan jaringan yang mengizinkan lalu lintas jaringan yang diperlukan antara KDC dapat diterima.

Opsional, setelah Anda membuat kepercayaan lintas ranah dengan Direktori Aktif menggunakan KDC pada satu klaster, Anda dapat membuat klaster lain menggunakan konfigurasi keamanan yang berbeda untuk referensi KDC pada klaster pertama sebagai KDC eksternal. Untuk konfigurasi keamanan dan pengaturan klaster contoh, lihat KDC klaster eksternal dengan kepercayaan lintas ranah Direktori Aktif.

Untuk informasi lebih lanjut tentang support Amazon EMR untuk Kerberos dan KDC, serta tautan ke Dokumentasi MIT Kerberos, lihat Gunakan Kerberos untuk otentikasi dengan Amazon EMR.

> **Important**
>
> Amazon EMR tidak support kepercayaan lintas ranah dengan AWS Directory Service for Microsoft Active Directory.

Langkah 1: Mengatur VPC dan subnet

Langkah 2: Peluncuran dan menginstal pengendali domain Direktori Aktif

Langkah 3: Tambahkan akun ke domain untuk EMR Cluster

Langkah 4: Konfigurasi kepercayaan masuk pada pengendali domain Direktori Aktif

Langkah 5: Gunakan opsi DHCP yang ditetapkan untuk menentukan pengendali domain Direktori Aktif sebagai server DNS VPC

Langkah 6: Meluncurkan klaster EMR Kerberized

Langkah 7: Buat pengguna HDFS dan atur izin pada cluster untuk akun Active Directory

Langkah 1: Mengatur VPC dan subnet

Langkah-langkah berikut menunjukkan menciptakan VPC dan subnet sehingga KDC klaster khusus dapat mencapai pengendali domain Direktori Aktif dan menyelesaikan nama domain. Di langkah-langkah ini, resolusi nama domain disediakan oleh referensi pengendali domain Direktori Aktif sebagai server nama domain di DHCP pilihan ditetapkan. Untuk informasi selengkapnya, lihat Langkah 5: Gunakan opsi DHCP yang ditetapkan untuk menentukan pengendali domain Direktori Aktif sebagai server DNS VPC.

Pengendali domain KDC dan Direktori Aktif harus mampu menyelesaikan nama domain satu sama lain. Hal ini memungkinkan Amazon EMR untuk bergabung dengan komputer ke domain dan secara otomatis mengkonfigurasi akun Linux yang sesuai dan parameter SSH pada instance cluster.

Jika Amazon EMR tidak dapat menyelesaikan nama domain, Anda dapat membuat referensi pada kepercayaan menggunakan alamat IP pengendali domain Direktori Aktif. Namun, Anda harus menambahkan akun Linux secara manual, menambahkan prinsip yang sesuai ke KDC khusus cluster, dan mengkonfigurasi SSH.

Untuk mengatur VPC dan subnet

1. Buat Amazon VPC dengan subnet publik tunggal Untuk informasi selengkapnya, lihat Langkah 1: Buat VPC di Panduan Memulai Amazon VPC.

   > **Important**
   >
   > Ketika Anda menggunakan pengendali domain Direktori Aktif Microsoftmemilih blok CIDR untuk klaster EMR sehingga semua panjang alamat IPv4 kurang dari sembilan karakter (misalnya, 10.0.0.0/16). Hal ini karena nama DNS klaster komputer yang digunakan ketika komputer bergabung dengan direktori Direktori Aktif. AWS menugaskan

Nama host DNS berdasarkan alamat IPv4 yang membuat alamat IP membuat nama DNS menjadi lebih dari 15 karakter. Direktori Aktif memiliki batas 15 karakter untuk mendaftar dan bergabung dengan nama komputer, dan memotong nama yang lebih panjang, yang dapat menyebabkan kesalahan tak terduga.

2. Menghapus opsi default DHCP yang ditetapkan untuk VPC. Untuk informasi selengkapnya, lihat Mengubah VPC untuk menggunakan opsi tidak menggunakan DHCP. Kemudian, Anda menambahkan yang baru yang menentukan pengendali domain Direktori Aktif sebagai server DNS.
3. Mengonfirmasi bahwa support DNS diaktifkan untuk VPC, yaitu Hostnames DNS dan Resolusi DNS keduanya diaktifkan. Mereka diaktifkan secara default. Untuk informasi selengkapnya, lihat Memperbarui support DNS untuk VPC Anda.
4. Mengonfirmasi bahwa VPC Anda memiliki gateway internet terlampir, yang merupakan default. Untuk informasi selengkapnya, lihat Membuat dan melampirkan gateway internet.

> **Note**
>
> Gateway internet digunakan di contoh ini karena Anda membuat pengendali domain baru untuk VPC. Gateway internet mungkin tidak diperlukan untuk aplikasi Anda. Satu-satunya persyaratan adalah bahwa KDC khusus klaster dapat mengakses pengendali domain Direktori Aktif.

5. Membuat tabel rute kustom, menambahkan rute yang menargetkan Gateway Internet, dan versi terbaru melampirkannya ke subnet Anda. Untuk informasi selengkapnya, lihat Buat tabel rute kustom.
6. Ketika Anda meluncurkan instans EC2 untuk pengendali domain, itu harus memiliki alamat IPv4 publik statis bagi Anda untuk connect dalam menggunakan RDP. Cara termudah untuk melakukannya adalah mengonfigurasi subnet Anda untuk menetapkan alamat IPv4 publik secara otomatis. Ini bukan pengaturan default ketika subnet dibuat. Untuk informasi selengkapnya, lihat Memodifikasi atribut pengalamatan IPv4 publik untuk subnet Anda. Opsional, Anda dapat menetapkan alamat saat Anda meluncurkan instans tersebut. Untuk informasi selengkapnya, lihat Menugaskan alamat IPv4 publik selama peluncuran instans.
7. Setelah selesai, buat catatan VPC dan subnet ID Anda. Anda menggunakannya nanti ketika Anda meluncurkan pengendali domain Direktori Aktif dan klaster.

Langkah 2: Peluncuran dan menginstal pengendali domain Direktori Aktif

1. Peluncuran instans EC2 berdasarkan Microsoft Windows Server 2016 Base AMI. Kami merekomendasikan m4.xlarge atau tipe instans yang lebih baik. Untuk informasi selengkapnya, lihat Meluncurkan AWS Marketplace instans di Panduan Pengguna Amazon EC2 untuk Instans Windows.
2. Membuat catatan ID grup dari grup keamanan yang terkait dengan instans EC2. Anda membutuhkannya untuk Langkah 6: Meluncurkan klaster EMR Kerberized. Kami menggunakan *sg-012xrlmdomain345*. Atau, Anda dapat menentukan grup keamanan yang berbeda untuk klaster EMR dan instans ini yang mengizinkan lalu lintas antara mereka. Untuk informasi selengkapnya, lihat Grup Keamanan Amazon EC2 untuk instans Linux di Panduan Pengguna Amazon EC2 untuk Instans Linux.
3. Connect ke instans EC2 menggunakan RDP. Untuk informasi selengkapnya, lihat Connect ke instans Windows Anda di Panduan Pengguna Amazon EC2 untuk Instans Windows .
4. Mulai Pengelola Server untuk menginstal dan mengonfigurasi peran Layanan domain Direktori Aktif di server. Promosikan server ke pengendali domain dan tugaskan nama domain (contoh yang kita gunakan di sini adalah *ad.domain.com*). Membuat catatan nama domain karena Anda memerlukannya nanti ketika Anda membuat konfigurasi keamanan EMR dan klaster. Jika Anda baru dalam hal menyiapkan Direktori Aktif, Anda dapat mengikuti petunjuk di Cara mengatur Direktori Aktif (AD) di Windows Server 2016.

   Instans me-restart setelah Anda selesai.

Langkah 3: Tambahkan akun ke domain untuk EMR Cluster

RDP ke pengontrol domain Active Directory untuk membuat akun di Pengguna Direktori Aktif dan Komputer untuk setiap pengguna cluster. Untuk selengkapnya, lihat Membuat Akun Pengguna di Pengguna Direktori Aktif dan Komputer di situs Microsoft Learn. Catat setiap Nama logon pengguna pengguna. Anda akan memerlukan ini ketika Anda mengonfigurasi klaster.

Selain itu, buat akun dengan hak istimewa yang cukup untuk bergabung dengan komputer ke domain. Anda menentukan akun ini ketika Anda membuat sebuah klaster. Amazon EMR menggunakannya untuk menggabungkan instans klaster untuk domain. Anda menentukan akun ini dan kata sandinya di Langkah 6: Meluncurkan klaster EMR Kerberized. Untuk mendelegasikan hak istimewa bergabung komputer ke akun, kami sarankan Anda membuat grup dengan hak istimewa bergabung dan kemudian menetapkan pengguna ke grup. Untuk instruksi, lihat Mendelegasikan hak istimwwa bergabung direktori di AWS Directory Service Panduan Administrasi.

Langkah 4: Konfigurasi kepercayaan masuk pada pengendali domain Direktori Aktif

Perintah contoh di bawah ini membuat kepercayaan di Direktori Aktif, yang merupakan satu arah, masuk, non-transitif, kepercayaan ranah dengan KDC khusus klaster. Contoh yang kita gunakan untuk ranah klaster adalah *EC2.INTERNAL*. Ganti *KDC-FQDN dengan nama DNS* Publik yang terdaftar untuk simpul utama Amazon EMR yang menghosting KDC. Parameter `passwordt` menentukan kata sandi utama lintas ranah, yang Anda tentukan bersama dengan ranah klaster saat Anda membuat klaster. Nama ranah berasal dari nama domain default di `us-east-1` untuk klaster. `Domain` adalah domain Direktori Aktif di mana Anda menciptakan kepercayaan, yang merupakan kasus yang lebih kecil oleh konvensi. Contoh menggunakan *ad.domain.com*

Buka prompt perintah Windows dengan hak istimewa administrator dan ketik perintah berikut untuk membuat hubungan kepercayaan pada pengendali domain Direktori Aktif:

```
C:\Users\Administrator> ksetup /addkdc EC2.INTERNAL KDC-FQDN
C:\Users\Administrator> netdom trust EC2.INTERNAL /Domain:ad.domain.com /add /realm /
passwordt:MyVeryStrongPassword
C:\Users\Administrator> ksetup /SetEncTypeAttr EC2.INTERNAL AES256-CTS-HMAC-SHA1-96
```

Langkah 5: Gunakan opsi DHCP yang ditetapkan untuk menentukan pengendali domain Direktori Aktif sebagai server DNS VPC

Sekarang bahwa pengendali domain Direktori Aktif dikonfigurasi, Anda harus mengonfigurasi VPC untuk menggunakannya sebagai server nama domain untuk resolusi nama di VPC Anda. Untuk melakukannya, lampirkan set opsi DHCP. Tentukan Nama domain sebagai nama domain klaster Anda - misalnya, `ec2.internal` jika klaster Anda berada di us-east-1 atau *region*`.compute.internal` untuk wilayah lain. Untuk server nama Domain, Anda harus menentukan alamat IP pengontrol domain Active Directory (yang harus dapat dijangkau dari cluster) sebagai entri pertama, diikuti oleh AmazonProvidedDNS (misalnya, ***xx.xx.xx.xx***, DNS). AmazonProvided Untuk informasi selengkapnya, lihat Mengganti set opsi DHCP.

Langkah 6: Meluncurkan klaster EMR Kerberized

1. Di Amazon EMR, buat konfigurasi keamanan yang menentukan pengendali domain Direktori Aktif yang Anda buat di langkah-langkah sebelumnya. Perintah contoh ditunjukkan di bawah ini. Ganti domain, *ad.domain.com*, dengan nama domain yang Anda tentukan di Langkah 2: Peluncuran dan menginstal pengendali domain Direktori Aktif.

   ```
   aws emr create-security-configuration --name MyKerberosConfig \
   ```

```
--security-configuration '{
  "AuthenticationConfiguration": {
    "KerberosConfiguration": {
      "Provider": "ClusterDedicatedKdc",
      "ClusterDedicatedKdcConfiguration": {
        "TicketLifetimeInHours": 24,
        "CrossRealmTrustConfiguration": {
          "Realm": "AD.DOMAIN.COM",
          "Domain": "ad.domain.com",
          "AdminServer": "ad.domain.com",
          "KdcServer": "ad.domain.com"
        }
      }
    }
  }
}'
```

2. Buat klaster dengan atribut berikut:

   - Gunakan opsi `--security-configuration` untuk menentukan konfigurasi keamanan yang Anda buat. Kami gunakan *`MyKerberosConfig`*dalam contoh.
   - Gunakan properti `SubnetId` dari `--ec2-attributes option` untuk menentukan subnet yang Anda buat di Langkah 1: Mengatur VPC dan subnet. Kita menggunakan *`step1-subnet`* di contoh.
   - Gunakan `AdditionalMasterSecurityGroups` dan `AdditionalSlaveSecurityGroups` `--ec2-attributes` opsi untuk menentukan bahwa grup keamanan yang terkait dengan pengontrol domain AD dari Langkah 2: Peluncuran dan menginstal pengendali domain Direktori Aktif dikaitkan dengan simpul utama klaster serta node inti dan tugas. Kami menggunakan *`sg-012xrlmdomain345`* di contoh.

   Gunakan `--kerberos-attributes` untuk menentukan atribut Kerberos khusus klaster berikut:

   - Ranah untuk klaster yang Anda tentukan ketika Anda mengatur pengendali domain Direktori Aktif.
   - Kata sandi utama kepercayaan lintas ranah yang Anda tentukan sebagai `passwordt` di Langkah 4: Konfigurasi kepercayaan masuk pada pengendali domain Direktori Aktif.
   - `KdcAdminPassword`, yang dapat Anda gunakan untuk mengelola KDC khusus klaster.
   - Nama logon dan kata sandi pengguna akun Direktori Aktif dengan hak istimewa gabungan komputer yang Anda buat di Langkah 3: Tambahkan akun ke domain untuk EMR Cluster.

   Contoh berikut meluncurkan klaster Kerberized.

```
aws emr create-cluster --name "MyKerberosCluster" \
--release-label emr-5.10.0 \
--instance-type m5.xlarge \
--instance-count 3 \
--ec2-attributes InstanceProfile=EMR_EC2_DefaultRole,KeyName=MyEC2KeyPair,\
SubnetId=step1-subnet, AdditionalMasterSecurityGroups=sg-012xrlmdomain345,
AdditionalSlaveSecurityGroups=sg-012xrlmdomain345\
--service-role EMR_DefaultRole \
--security-configuration MyKerberosConfig \
--applications Name=Hadoop Name=Hive Name=Oozie Name=Hue Name=HCatalog Name=Spark \
--kerberos-attributes Realm=EC2.INTERNAL,\
KdcAdminPassword=MyClusterKDCAdminPwd,\
ADDomainJoinUser=ADUserLogonName,ADDomainJoinPassword=ADUserPassword,\
CrossRealmTrustPrincipalPassword=MatchADTrustPwd
```

Langkah 7: Buat pengguna HDFS dan atur izin pada cluster untuk akun Active Directory

Saat menyiapkan hubungan kepercayaan dengan Active Directory, Amazon EMR membuat pengguna Linux di cluster untuk setiap akun Active Directory. Misalnya, nama logon pengguna `LiJuan` di Active Directory memiliki akun Linux. `lijuan` Nama pengguna Direktori Aktif dapat berisi huruf besar, tetapi Linux tidak menerima casing Direktori Aktif.

Untuk memungkinkan pengguna Anda masuk ke cluster untuk menjalankan pekerjaan Hadoop, Anda harus menambahkan direktori pengguna HDFS untuk akun Linux mereka, dan memberikan setiap pengguna kepemilikan direktori mereka. Untuk melakukannya, kami merekomendasikan Anda menjalankan script yang disimpan ke Amazon S3 sebagai langkah klaster. Atau, Anda dapat menjalankan perintah dalam skrip di bawah ini dari baris perintah pada node utama. Gunakan key pair EC2 yang Anda tentukan ketika Anda membuat cluster untuk terhubung ke node utama melalui SSH sebagai pengguna Hadoop. Untuk informasi selengkapnya, lihat Menggunakan key pair EC2 untuk kredensi SSH.

Jalankan perintah berikut untuk menambahkan langkah ke klaster yang menjalankan script, *AddHDFSUsers.sh*.

```
aws emr add-steps --cluster-id <j-2AL4XXXXXX5T9> \
--steps Type=CUSTOM_JAR,Name=CustomJAR,ActionOnFailure=CONTINUE,\
Jar=s3://region.elasticmapreduce/libs/script-runner/script-runner.jar,Args=["s3://DOC-
EXAMPLE-BUCKET/AddHDFSUsers.sh"]
```

Isi file *AddHDFSUsers.sh* adalah sebagai berikut.

```
#!/bin/bash
# AddHDFSUsers.sh script

# Initialize an array of user names from AD or Linux users and KDC principals created
 manually on the cluster
ADUSERS=("lijuan" "marymajor" "richardroe" "myusername")

# For each user listed, create an HDFS user directory
# and change ownership to the user

for username in ${ADUSERS[@]}; do
     hdfs dfs -mkdir /user/$username
     hdfs dfs -chown $username:$username /user/$username
done
```

Grup Direktori Aktif dipetakan ke grup Hadoop

Amazon EMR menggunakan System Security Services Daemon (SSD) untuk memetakan grup Direktori Aktif untuk grup Hadoop. Untuk mengonfirmasi pemetaan grup, setelah Anda masuk ke node utama seperti yang dijelaskanMenggunakan SSH untuk connect ke klaster Kerberized, Anda dapat menggunakan `hdfs groups` perintah untuk mengonfirmasi bahwa grup Active Directory yang menjadi milik akun Active Directory Anda telah dipetakan ke grup Hadoop untuk pengguna Hadoop yang sesuai di cluster. Anda juga dapat memeriksa pemetaan grup pengguna lain dengan menentukan satu nama pengguna atau lebih dengan perintah, misalnya `hdfs groups` *`lijuan`*. Untuk informasi selengkapnya, lihat grup di Panduan Perintah HDFS Apache.

Gunakan Active Directory atau server LDAP untuk otentikasi dengan Amazon EMR

Dengan Amazon EMR merilis 6.12.0 dan yang lebih tinggi, Anda dapat menggunakan protokol LDAP over SSL (LDAPS) untuk meluncurkan cluster yang terintegrasi secara native dengan server identitas perusahaan Anda. LDAP (Lightweight Directory Access Protocol) adalah protokol aplikasi terbuka dan netral vendor yang mengakses dan memelihara data. LDAP umumnya digunakan untuk otentikasi pengguna terhadap server identitas perusahaan yang di-host pada aplikasi seperti Active Directory (AD) dan OpenLDAP. Dengan integrasi asli ini, Anda dapat menggunakan server LDAP Anda untuk mengautentikasi pengguna di Amazon EMR.

Sorotan integrasi Amazon EMR LDAP meliputi:

- Amazon EMR mengonfigurasi aplikasi yang didukung untuk mengautentikasi dengan otentikasi LDAP atas nama Anda.
- Amazon EMR mengonfigurasi dan memelihara keamanan untuk aplikasi yang didukung dengan protokol Kerberos. Anda tidak perlu memasukkan perintah atau skrip apa pun.
- Anda mendapatkan kontrol akses halus (FGAC) melalui otorisasi Apache Ranger untuk database dan tabel Hive Metastore. Lihat Mengintegrasikan Amazon EMR dengan Apache Ranger untuk informasi selengkapnya.
- Ketika Anda memerlukan kredensyal LDAP untuk mengakses kluster, Anda mendapatkan kontrol akses halus (FGAC) atas siapa yang dapat mengakses kluster EMR Anda melalui SSH.

Halaman-halaman berikut memberikan gambaran konseptual, prasyarat, dan langkah-langkah untuk meluncurkan cluster EMR dengan integrasi Amazon EMR LDAP.

Topik

Ikhtisar LDAP dengan Amazon EMR

Lightweight Directory Access Protocol (LDAP) adalah protokol perangkat lunak yang digunakan administrator jaringan untuk mengelola dan mengontrol akses ke data dengan mengautentikasi pengguna dalam jaringan perusahaan. Protokol LDAP menyimpan informasi dalam hierarkis, struktur direktori pohon. Untuk informasi lebih lanjut, lihat Konsep LDAP Dasar di LDAP.com.

Dalam jaringan perusahaan, banyak aplikasi mungkin menggunakan protokol LDAP untuk mengautentikasi pengguna. Dengan integrasi Amazon EMR LDAP, kluster EMR secara native dapat menggunakan protokol LDAP yang sama dengan konfigurasi keamanan tambahan.

Ada dua implementasi utama protokol LDAP yang didukung Amazon EMR: Active Directory dan OpenLDAP. Sementara implementasi lain dimungkinkan, sebagian besar sesuai dengan protokol otentikasi yang sama seperti Active Directory atau OpenLDAP.

Direktori Aktif (AD)

Active Directory (AD) adalah layanan direktori dari Microsoft untuk jaringan domain Windows. AD disertakan pada sebagian besar sistem operasi Windows Server, dan dapat berkomunikasi dengan klien melalui protokol LDAP dan LDAPS. Untuk autentikasi, Amazon EMR mencoba mengikat pengguna dengan instans AD Anda dengan Nama Utama Pengguna (UPN) sebagai nama dan kata sandi yang dibedakan. UPN menggunakan format `username@domain_name` standar.

OpenLDAP

OpenLDAP adalah implementasi open source gratis dari protokol LDAP. Untuk autentikasi, Amazon EMR mencoba mengikat pengguna dengan instans OpenLDAP Anda dengan nama domain yang memenuhi syarat (FQDN) sebagai nama dan kata sandi yang dibedakan. FQDN menggunakan format standar. `username_attribute=username,LDAP_user_search_base` Umumnya, `username_attribute` nilainya`uid`, dan `LDAP_user_search_base` nilainya berisi atribut pohon yang mengarah ke pengguna. Sebagai contoh, `ou=People,dc=example,dc=com`.

Implementasi gratis dan open-source lainnya dari protokol LDAP biasanya mengikuti FQDN yang serupa dengan OpenLDAP untuk nama-nama terkemuka penggunanya.

Komponen LDAP untuk Amazon EMR

Anda dapat menggunakan server LDAP Anda untuk mengautentikasi dengan Amazon EMR dan aplikasi apa pun yang langsung digunakan pengguna pada cluster EMR melalui komponen-komponen berikut.

Agen Rahasia

Agen Rahasia adalah proses on-cluster yang mengautentikasi semua permintaan pengguna. Agen Rahasia membuat pengguna mengikat ke server LDAP Anda atas nama aplikasi yang didukung pada cluster EMR. Agen Rahasia berjalan sebagai `emrsecretagent` pengguna, dan menulis log ke `/emr/secretagent/log` direktori. Log ini memberikan rincian tentang status permintaan otentikasi setiap pengguna dan kesalahan apa pun yang mungkin muncul selama otentikasi pengguna.

Layanan Keamanan Sistem Daemon (SSSD)

SSSD adalah daemon yang berjalan pada setiap node dari cluster EMR berkemampuan LDAP. SSSD membuat dan mengelola pengguna UNIX untuk menyinkronkan identitas perusahaan jarak jauh Anda ke setiap node. Aplikasi berbasis benang seperti Hive dan Spark mengharuskan pengguna UNIX lokal ada di setiap node yang menjalankan kueri untuk pengguna.

Dukungan aplikasi dan pertimbangan dengan LDAP untuk Amazon EMR

Aplikasi yang didukung dengan LDAP untuk Amazon EMR

> **Important**
>
> Aplikasi yang tercantum di halaman ini adalah satu-satunya aplikasi yang didukung Amazon EMR untuk LDAP. Untuk memastikan keamanan klaster, Anda hanya dapat menyertakan aplikasi yang kompatibel dengan LDAP saat membuat klaster EMR dengan LDAP diaktifkan. Jika Anda mencoba menginstal aplikasi lain yang tidak didukung, Amazon EMR akan menolak permintaan Anda untuk klaster baru.

Amazon EMR merilis 6.12 dan lebih tinggi mendukung integrasi LDAP dengan aplikasi berikut:

- Apache Livy
- Sarang Apache hingga HiveServer 2 (HS2)
- Trino
- Presto
- Hue

Anda juga dapat menginstal aplikasi berikut pada cluster EMR dan mengonfigurasinya untuk memenuhi kebutuhan keamanan Anda:

- Apache Spark
- Apache Hadoop

Fitur yang didukung dengan LDAP untuk Amazon EMR

Anda dapat menggunakan fitur EMR Amazon berikut dengan integrasi LDAP:

> **Note**
>
> Untuk menjaga kredensyal LDAP tetap aman, Anda harus menggunakan enkripsi dalam transit untuk mengamankan aliran data di dalam dan di luar klaster. Untuk informasi selengkapnya tentang enkripsi dalam perjalanan, lihatEnkripsi data at rest dan dalam transit.

- Enkripsi dalam perjalanan (wajib) dan saat istirahat
- Grup klaster, armada instans, dan instans Spot
- Konfigurasi ulang aplikasi pada klaster berjalan
- Server-side encryption (SSE) EMRFS

Fitur yang tidak didukung

Pertimbangkan batasan berikut saat Anda menggunakan integrasi Amazon EMR LDAP:

- Amazon EMR menonaktifkan langkah-langkah untuk cluster dengan LDAP diaktifkan.
- Amazon EMR tidak mendukung peran runtime dan AWS Lake Formation integrasi untuk cluster dengan LDAP diaktifkan.
- Amazon EMR tidak mendukung LDAP dengan StartTLS.
- Amazon EMR tidak mendukung mode ketersediaan tinggi (cluster dengan beberapa node utama) untuk cluster dengan LDAP diaktifkan.
- Anda tidak dapat memutar kredensyal atau sertifikat bind untuk klaster dengan LDAP diaktifkan. Jika salah satu bidang tersebut diputar, sebaiknya Anda memulai klaster baru dengan kredensyal atau sertifikat bind yang diperbarui.
- Anda harus menggunakan basis pencarian yang tepat dengan LDAP. Basis pencarian pengguna dan grup LDAP tidak mendukung filter pencarian LDAP.

Konfigurasikan dan luncurkan cluster EMR dengan LDAP

Bagian ini mencakup cara mengkonfigurasi Amazon EMR untuk digunakan dengan otentikasi LDAP.

Topik

- Tambahkan AWS Secrets Manager izin ke peran instans EMR Amazon
- Buat konfigurasi keamanan Amazon EMR untuk integrasi LDAP
- Luncurkan cluster EMR yang mengautentikasi dengan LDAP

Tambahkan AWS Secrets Manager izin ke peran instans EMR Amazon

Amazon EMR menggunakan peran layanan IAM untuk melakukan tindakan atas nama Anda untuk menyediakan dan mengelola klaster. Peran layanan untuk instans EC2 cluster, juga disebut profil

instans EC2 untuk Amazon EMR, adalah jenis peran layanan khusus yang diberikan Amazon EMR ke setiap instans EC2 dalam klaster saat diluncurkan.

Untuk menentukan izin kluster EMR agar berinteraksi dengan data Amazon S3 dan layanan AWS lainnya, tentukan profil instans Amazon EC2 kustom, bukan saat Anda meluncurkan klaster. `EMR_EC2_DefaultRole` Untuk informasi selengkapnya, silakan lihat Peran layanan untuk instans EC2 klaster (profil instans EC2) dan Kustom IAM role.

Tambahkan pernyataan berikut ke profil instans EC2 default untuk memungkinkan Amazon EMR menandai sesi dan mengakses yang menyimpan sertifikat AWS Secrets Manager LDAP.

```
    {
      "Sid": "AllowAssumeOfRolesAndTagging",
      "Effect": "Allow",
      "Action": ["sts:TagSession", "sts:AssumeRole"],
      "Resource": [
        "arn:aws:iam::111122223333:role/LDAP_DATA_ACCESS_ROLE_NAME",
        "arn:aws:iam::111122223333:role/LDAP_USER_ACCESS_ROLE_NAME"
      ]
    },
    {
        "Sid": "AllowSecretsRetrieval",
        "Effect": "Allow",
        "Action": "secretsmanager:GetSecretValue",
        "Resource": [
            "arn:aws:secretsmanager:us-east-1:111122223333:secret:LDAP_SECRET_NAME*",
            "arn:aws:secretsmanager:us-
east-1:111122223333:secret:ADMIN_LDAP_SECRET_NAME*"
        ]
    }
```

> **Note**
>
> Permintaan klaster Anda akan gagal jika Anda lupa * karakter wildcard di akhir nama rahasia saat Anda menetapkan izin Secrets Manager. Wildcard mewakili versi rahasia.
> Anda juga harus membatasi cakupan AWS Secrets Manager kebijakan hanya pada sertifikat yang dibutuhkan klaster Anda untuk menyediakan instance.

Buat konfigurasi keamanan Amazon EMR untuk integrasi LDAP

Sebelum Anda dapat meluncurkan kluster EMR dengan integrasi LDAP, gunakan langkah-langkah Membuat konfigurasi keamanan untuk membuat konfigurasi keamanan EMR Amazon untuk klaster. Selesaikan konfigurasi berikut di `LDAPConfiguration` blok di bawah`AuthenticationConfiguration`, atau di bidang yang sesuai di bagian Konfigurasi Keamanan konsol EMR Amazon:

`EnableLDAPAuthentication`

Opsi konsol: Protokol otentikasi: LDAP

Untuk menggunakan integrasi LDAP, setel opsi ini ke `true` atau pilih sebagai protokol otentikasi Anda saat Anda membuat klaster di konsol. Secara default, `EnableLDAPAuthentication` adalah `true` saat Anda membuat konfigurasi keamanan di konsol EMR Amazon.

`LDAPServerURL`

Opsi konsol: Lokasi server LDAP

Lokasi server LDAP termasuk awalan:. `ldaps://`*`location_of_server`*

`BindCertificateARN`

Opsi konsol: Sertifikat SSL LDAP

AWS Secrets ManagerARN yang berisi sertifikat untuk menandatangani sertifikat SSL yang digunakan server LDAP. Jika server LDAP Anda ditandatangani oleh Public Certificate Authority (CA), Anda dapat memberikan AWS Secrets Manager ARN dengan file kosong. Untuk informasi selengkapnya tentang cara menyimpan sertifikat di Secrets Manager, lihatMenyimpan sertifikat TLS di AWS Secrets Manager.

`BindCredentialsARN`

Opsi konsol: Server LDAP mengikat kredensyal

AWS Secrets ManagerARN yang berisi pengguna admin LDAP mengikat kredensyal. Kredensyal disimpan sebagai objek JSON. Hanya ada satu pasangan kunci-nilai dalam rahasia ini; kunci dalam pasangan adalah nama pengguna, dan nilainya adalah kata sandi. Sebagai contoh, `{"uid=admin,cn=People,dc=example,dc=com": "AdminPassword1"}`. Ini adalah bidang opsional kecuali Anda mengaktifkan login SSH untuk cluster EMR Anda. Dalam banyak konfigurasi, instance Active Directory memerlukan kredensyal bind untuk memungkinkan SSSD menyinkronkan pengguna.

`LDAPAccessFilter`

Opsi konsol: Filter akses LDAP

Menentukan subset objek dalam server LDAP Anda yang dapat mengautentikasi. Misalnya, jika semua yang ingin Anda berikan akses ke semua pengguna dengan kelas `posixAccount` objek di server LDAP Anda, tentukan filter akses sebagai`(objectClass=posixAccount)`.

`LDAPUserSearchBase`

Opsi konsol: Basis pencarian pengguna LDAP

Basis pencarian yang dimiliki pengguna Anda di dalam server LDAP Anda. Sebagai contoh, `cn=People,dc=example,dc=com`.

`LDAPGroupSearchBase`

Opsi konsol: Basis pencarian grup LDAP

Basis pencarian yang dimiliki grup Anda di dalam server LDAP Anda. Sebagai contoh, `cn=Groups,dc=example,dc=com`.

`EnableSSHLogin`

Opsi konsol: Login SSH

Menentukan apakah atau tidak untuk mengizinkan otentikasi password dengan kredensyal LDAP. Kami tidak menyarankan Anda mengaktifkan opsi ini. Pasangan kunci adalah rute yang lebih aman untuk memungkinkan akses ke cluster EMR. Bidang ini opsional dan default ke. `false`

`LDAPServerType`

Opsi konsol: Jenis server LDAP

Menentukan jenis server LDAP yang terhubung dengan Amazon EMR. Opsi yang didukung adalah Active Directory dan OpenLDAP. Jenis server LDAP lainnya mungkin berfungsi, tetapi Amazon EMR tidak secara resmi mendukung jenis server lainnya. Untuk informasi selengkapnya, lihat Komponen LDAP untuk Amazon EMR.

`ActiveDirectoryConfigurations`

Sub-blok yang diperlukan untuk konfigurasi keamanan yang menggunakan jenis server Active Directory.

`ADDomain`

Opsi konsol: Domain Direktori Aktif

Nama domain yang digunakan untuk membuat User Principal Name (UPN) untuk otentikasi pengguna dengan konfigurasi keamanan yang menggunakan jenis server Active Directory.

Pertimbangan untuk konfigurasi keamanan dengan LDAP dan Amazon EMR

- Untuk membuat konfigurasi keamanan dengan integrasi Amazon EMR LDAP, Anda harus menggunakan enkripsi dalam perjalanan. Untuk informasi tentang enkripsi dalam perjalanan, lihatEnkripsi data at rest dan dalam transit.
- Anda tidak dapat menentukan konfigurasi Kerberos dalam konfigurasi keamanan yang sama. Amazon EMR menyediakan KDC thar didedikasikan untuk secara otomatis, dan mengelola kata sandi admin untuk KDC ini. Pengguna tidak dapat mengakses kata sandi admin ini.
- Anda tidak dapat menentukan peran runtime IAM dan AWS Lake Formation dalam konfigurasi keamanan yang sama.
- `LDAPServerURL`Harus memiliki `ldaps://` protokol dalam nilainya.
- Tidak `LDAPAccessFilter` bisa kosong.

Gunakan LDAP dengan integrasi Apache Ranger untuk Amazon EMR

Dengan integrasi LDAP untuk Amazon EMR, Anda dapat lebih berintegrasi dengan Apache Ranger. Saat Anda menarik pengguna LDAP Anda ke Ranger, Anda kemudian dapat mengaitkan pengguna tersebut dengan server kebijakan Apache Ranger untuk diintegrasikan dengan Amazon EMR dan aplikasi lainnya. Untuk melakukan ini, tentukan `RangerConfiguration` bidang `AuthorizationConfiguration` dalam konfigurasi keamanan yang Anda gunakan dengan klaster LDAP Anda. Untuk informasi selengkapnya tentang cara mengatur konfigurasi keamanan, lihatBuat konfigurasi keamanan EMR.

Saat Anda menggunakan LDAP dengan Amazon EMR, Anda tidak perlu menyediakan `KerberosConfiguration` integrasi EMR Amazon untuk Apache Ranger.

Luncurkan cluster EMR yang mengautentikasi dengan LDAP

Gunakan langkah-langkah berikut untuk meluncurkan cluster EMR dengan LDAP atau Active Directory.

1. Siapkan lingkungan Anda:
   - Pastikan node pada cluster EMR Anda dapat berkomunikasi dengan Amazon S3 dan. AWS Secrets Manager Untuk informasi selengkapnya tentang cara mengubah peran profil instans

EC2 Anda untuk berkomunikasi dengan layanan ini, lihatTambahkan AWS Secrets Manager izin ke peran instans EMR Amazon.

- Jika Anda berencana untuk menjalankan klaster EMR Anda di subnet pribadi, Anda harus menggunakan dan titik akhir AWS PrivateLink Amazon VPC, atau menggunakan transalasi alamat jaringan (NAT) untuk mengonfigurasi VPC agar berkomunikasi dengan S3 dan Secrets Manager. Untuk informasi selengkapnya, lihat AWS PrivateLinkdan titik akhir VPC dan instans NAT di Panduan Memulai VPC Amazon.
- Pastikan ada konektivitas jaringan antara cluster EMR Anda dan server LDAP. Cluster EMR Anda harus mengakses server LDAP Anda melalui jaringan. Node utama, inti, dan tugas untuk cluster berkomunikasi dengan server LDAP untuk menyinkronkan data pengguna. Jika server LDAP Anda berjalan di Amazon EC2, perbarui grup keamanan EC2 untuk menerima lalu lintas dari kluster EMR. Untuk informasi selengkapnya, lihat Tambahkan AWS Secrets Manager izin ke peran instans EMR Amazon.

2. Buat konfigurasi keamanan EMR Amazon untuk integrasi LDAP. Untuk informasi selengkapnya, lihat Buat konfigurasi keamanan Amazon EMR untuk integrasi LDAP.
3. Sekarang setelah Anda menyiapkan, gunakan langkah-langkah Meluncurkan klaster Amazon EMR untuk meluncurkan klaster Anda dengan konfigurasi berikut:

   - Pilih Amazon EMR rilis 6.12 atau lebih tinggi. Kami menyarankan Anda menggunakan rilis EMR Amazon terbaru.
   - Hanya tentukan atau pilih aplikasi untuk klaster Anda yang mendukung LDAP. Untuk daftar aplikasi yang didukung LDAP dengan Amazon EMR, lihat. Dukungan aplikasi dan pertimbangan dengan LDAP untuk Amazon EMR
   - Terapkan konfigurasi keamanan yang Anda buat di langkah sebelumnya.

Contoh menggunakan LDAP dengan Amazon EMR

Setelah Anda menyediakan kluster EMR yang menggunakan integrasi LDAP, Anda dapat memberikan kredensyal LDAP Anda ke aplikasi apa pun yang didukung melalui mekanisme autentikasi nama pengguna dan kata sandi bawaannya. Halaman ini menunjukkan beberapa contoh.

Menggunakan otentikasi LDAP dengan Apache Hive

Example - Sarang Apache

Contoh perintah berikut memulai sesi Apache Hive melalui HiveServer 2 dan Beeline:

```
beeline -u "jdbc:hive2://$HOSTNAME:10000/default;ssl=true;sslTrustStore=
$TRUSTSTORE_PATH;trustStorePassword=$TRUSTSTORE_PASS"  -n LDAP_USERNAME -
p LDAP_PASSWORD
```

Menggunakan otentikasi LDAP dengan Apache Livy

Example - Apache Livy

Contoh perintah berikut memulai sesi Livy melalui cURL. Ganti *ENCODED-KEYPAIR* dengan string yang dikodekan Base64 untuk. `username:password`

```
curl -X POST --data '{"proxyUser":"LDAP_USERNAME","kind": "pyspark"}' -H "Content-Type:
 application/json" -H "Authorization: Basic ENCODED-KEYPAIR" DNS_OF_PRIMARY_NODE:8998/
sessions
```

Menggunakan otentikasi LDAP dengan Presto

Example - Presto

Contoh perintah berikut memulai sesi Presto melalui CLI Presto:

```
presto-cli --user "LDAP_USERNAME" --password --catalog hive
```

Setelah Anda menjalankan perintah ini, masukkan kata sandi LDAP pada prompt.

Menggunakan otentikasi LDAP dengan Trino

Example - Trino

Contoh perintah berikut memulai sesi Trino melalui Trino CLI:

```
trino-cli --user "LDAP_USERNAME" --password --catalog hive
```

Setelah Anda menjalankan perintah ini, masukkan kata sandi LDAP pada prompt.

Menggunakan otentikasi LDAP dengan Hue

Anda dapat mengakses Hue UI melalui terowongan SSH yang Anda buat di cluster, atau Anda dapat mengatur server proxy untuk menyiarkan koneksi ke Hue secara publik. Karena Hue tidak

berjalan dalam mode HTTPS secara default, kami menyarankan Anda menggunakan lapisan enkripsi tambahan untuk memastikan bahwa komunikasi antara klien dan UI Hue dienkripsi dengan HTTPS. Ini mengurangi kemungkinan Anda secara tidak sengaja mengekspos kredensyal pengguna dalam teks biasa.

Untuk menggunakan UI Hue, buka UI Hue di browser Anda dan masukkan kata sandi nama pengguna LDAP Anda untuk masuk. Jika kredensialnya benar, Hue mencatat Anda dan menggunakan identitas Anda untuk mengautentikasi Anda dengan semua aplikasi yang didukung.

Menggunakan SSH untuk otentikasi kata sandi dan tiket Kerberos untuk aplikasi lain

> **Important**
>
> Kami tidak menyarankan Anda menggunakan otentikasi kata sandi ke SSH ke dalam cluster EMR.

Anda dapat menggunakan kredensyal LDAP Anda ke SSH ke cluster EMR. Untuk melakukan ini, atur `EnableSSHLogin` konfigurasi ke `true` dalam konfigurasi keamanan Amazon EMR yang Anda gunakan untuk memulai cluster. Kemudian, gunakan perintah berikut untuk SSH ke cluster setelah diluncurkan:

```
ssh username@EMR_PRIMARY_DNS_NAME
```

Setelah Anda menjalankan perintah ini, masukkan kata sandi LDAP pada prompt.

Amazon EMR menyertakan skrip on-cluster yang memungkinkan pengguna membuat file dan tiket tab Kerberos untuk digunakan dengan aplikasi yang didukung yang tidak menerima kredensyal LDAP secara langsung. Beberapa aplikasi ini termasuk`spark-submit`, Spark SQL, dan. PySpark

Jalankan `ldap-kinit` dan ikuti petunjuknya. Jika otentikasi berhasil, file tab Kerberos muncul di direktori home Anda dengan tiket Kerberos yang valid. Gunakan tiket Kerberos untuk menjalankan aplikasi seperti yang Anda lakukan di lingkungan Kerberized apa pun.

Integrasi Amazon EMR dengan AWS IAM Identity Center

Dengan Amazon EMR rilis 6.15.0 dan yang lebih tinggi, Anda dapat menggunakan identitas dari untuk AWS IAM Identity Center mengautentikasi dengan kluster EMR Amazon. Bagian berikut

memberikan gambaran konseptual, prasyarat, dan langkah-langkah yang diperlukan untuk meluncurkan cluster EMR dengan integrasi Identity Center.

Topik

Gambaran Umum

Penyebaran identitas tepercaya melalui IAM Identity Center dapat membantu Anda membuat atau menghubungkan identitas tenaga kerja Anda dengan aman, dan mengelola akses mereka secara terpusat di seluruh akun dan aplikasi. AWS Dengan kemampuan ini, pengguna dapat masuk ke aplikasi yang menggunakan propagasi identitas tepercaya, dan aplikasi itu dapat meneruskan identitas pengguna dalam permintaan yang dibuatnya untuk mengakses data dalam AWS layanan yang juga menggunakan propagasi identitas tepercaya. Karena akses dikelola berdasarkan identitas pengguna, pengguna tidak perlu menggunakan kredensyal pengguna lokal database atau mengambil peran IAM untuk mengakses data.

Identity Center adalah pendekatan yang direkomendasikan untuk otentikasi dan otorisasi tenaga kerja AWS untuk organisasi dari berbagai ukuran dan jenis. Dengan Identity Center, Anda dapat membuat dan mengelola identitas pengguna diAWS, atau menghubungkan sumber identitas yang ada, termasuk Microsoft Active Directory, Okta, Ping Identity, Google Workspace JumpCloud, dan Microsoft Entra ID (sebelumnya Azure AD).

Untuk informasi lebih lanjut, lihat Apa ituAWS IAM Identity Center? dan propagasi identitas tepercaya di seluruh aplikasi di Panduan AWS IAM Identity Center Pengguna.

Fitur dan manfaat

Integrasi Amazon EMR dengan IAM Identity Center memberikan manfaat berikut:

- Amazon EMR menyediakan kredensyal untuk menyampaikan Identitas Pusat Identitas Anda ke kluster EMR.
- Amazon EMR mengonfigurasi semua aplikasi yang didukung untuk mengautentikasi dengan kredensyal cluster.

- Amazon EMR mengonfigurasi dan memelihara keamanan aplikasi yang didukung dengan protokol Kerberos dan tidak ada perintah atau skrip yang diperlukan oleh Anda.
- Kemampuan untuk menerapkan otorisasi tingkat awalan Amazon S3 dengan identitas Pusat Identitas pada awalan S3 yang dikelola S3 S3 Access Grants.
- Kemampuan untuk menegakkan otorisasi tingkat tabel dengan identitas Pusat Identitas pada AWS Lake Formation tabel Glue yang dikelola. AWS

Memulai AWS IAM Identity Center integrasi untuk Amazon EMR

Bagian ini membantu Anda mengonfigurasi EMR Amazon untuk diintegrasikan. AWS IAM Identity Center

Topik

- Buat instance Pusat Identitas
- Buat peran IAM untuk Identity Center
- Membuat konfigurasi keamanan yang diaktifkan Pusat Identitas
- Membuat dan meluncurkan kluster yang diaktifkan Pusat Identitas
- Konfigurasikan Lake Formation untuk kluster EMR yang diaktifkan Pusat Identitas IAM
- Bekerja dengan Hibah Akses S3 pada kluster EMR yang diaktifkan Pusat Identitas IAM

Buat instance Pusat Identitas

Jika Anda belum memilikinya, buat instance Pusat Identitas di Wilayah AWS tempat Anda ingin meluncurkan cluster EMR Anda. Instance Pusat Identitas hanya dapat ada di satu Wilayah untuk sebuahAkun AWS.

Gunakan AWS CLI perintah berikut untuk membuat instance baru bernama*MyInstance*:

```
aws sso create-instance --name MyInstance
```

Buat peran IAM untuk Identity Center

Untuk mengintegrasikan Amazon EMRAWS IAM Identity Center, buat peran IAM yang mengautentikasi dengan Identity Center dari cluster EMR. Di bawah tenda, Amazon EMR

menggunakan SigV4 kredensyal untuk menyampaikan identitas Pusat Identitas ke layanan hilir seperti. AWS Lake Formation Peran Anda juga harus memiliki izin masing-masing untuk memanggil layanan hilir.

Saat Anda membuat peran, gunakan kebijakan izin berikut:

```
"Statement": [
 { // For IdC interaction
   "Sid": "IdCPermissions",
   "Effect": "Allow",
   "Action": [
      "sso-oauth:*"
   ],
   "Resource" : "*"
 },
 { // For Lake Formation and Glue interaction
   "Sid": "GlueandLakePermissions"
   "Effect": "Allow",
   "Action": [
    "glue:*",
    "lakeformation:GetDataAccess"
   ],
   "Resource" : "*"
 },
 { // For S3 Access Grants interaction
   "Sid": "StaircasePermissions",
   "Effect": "Allow",
   "Action": [
      "s3:GetDataAccess",
      "s3:GetAccessGrantsInstanceForPrefix"
   ]
   "Resource" : "*"
 }
]
```

Kebijakan kepercayaan untuk peran ini memungkinkan InstanceProfile peran untuk membiarkannya mengambil peran.

```
{
   "Sid": "AssumeRole",
   "Effect": "Allow",
   "Principal": {
```

```
        "AWS": "arn:aws:iam::12345678912:role/EMR_EC2_DefaultRole"
    },
    "Action": [
               "sts:AssumeRole",
                "sts:SetContext"
    ]
 }
```

Membuat konfigurasi keamanan yang diaktifkan Pusat Identitas

Untuk meluncurkan cluster EMR dengan integrasi IAM Identity Center, gunakan perintah contoh berikut untuk membuat konfigurasi keamanan Amazon EMR yang mengaktifkan Pusat Identitas. Setiap konfigurasi dijelaskan di bawah ini.

```
aws emr create-security-configuration --name "IdentityCenterConfiguration-with-lf-
Staircase-KC" --region "us-west-2" --endpoint-url https://elasticmapreduce-preprod.us-
west-2.amazonaws.com --security-configuration '{
 "AuthenticationConfiguration":{
  "IdentityCenterConfiguration":{
   "EnableIdentityCenter":true,
   "IdentityCenterApplicationAssigmentRequired":false,
   "IdentityCenterInstanceARN": "arn:aws:sso:::instance/ssoins-123xxxxxxxxx789",
   "IAMRoleForEMRIdentityCenterApplicationARN": "arn:aws:iam::123456789012:role/tip-
role"
     }
 },
 "AuthorizationConfiguration": {
  "LakeFormationConfiguration": {
   "EnableLakeFormation": true
  }
 },
 "EncryptionConfiguration": {
  "EnableInTransitEncryption": true,
  "EnableAtRestEncryption": false,
  "InTransitEncryptionConfiguration": {
   "TLSCertificateConfiguration": {
    "CertificateProviderType": "PEM",
    "S3Object": "s3://my-bucket/cert/my-certs.zip"
   }
  }
 }
}'
```

- **`EnableIdentityCenter`**— (wajib) Memungkinkan integrasi Pusat Identitas.
- **`IdentityCenterApplicationARN`**— (wajib) Pusat Identitas misalnya ARN.
- **`IAMRoleForEMRIdentityCenterApplicationARN`**— (wajib) Peran IAM yang mendapatkan token Identity Center dari cluster.
- **`IdentityCenterApplicationAssignmentRequired`** — (boolean) Mengatur jika tugas akan diperlukan untuk menggunakan aplikasi Pusat Identitas. Nilai default-nya adalah `true`.
- **`AuthenticationConfiguration`/`LakeFormationConfiguration`**— Secara opsional, konfigurasikan otentikasi:
  - **`EnableLakeFormation`**— Aktifkan otorisasi Lake Formation di cluster.

Untuk mengaktifkan integrasi Pusat Identitas dengan Amazon EMR, Anda harus menentukan `EncryptionConfiguration` dan. `IntransitEncryptionConfiguration`

Membuat dan meluncurkan kluster yang diaktifkan Pusat Identitas

Sekarang setelah Anda menyiapkan peran IAM yang mengautentikasi dengan Identity Center, dan membuat konfigurasi keamanan Amazon EMR yang mengaktifkan Pusat Identitas, Anda dapat membuat dan meluncurkan cluster sadar identitas Anda. Untuk langkah-langkah untuk meluncurkan klaster Anda dengan konfigurasi keamanan yang diperlukan, lihatMenentukan konfigurasi keamanan untuk sebuah klaster.

Secara opsional, lihat bagian berikut jika Anda ingin menggunakan kluster yang diaktifkan Pusat Identitas dengan opsi keamanan lain yang didukung Amazon EMR:

- Bekerja dengan Hibah Akses S3 pada kluster EMR yang diaktifkan Pusat Identitas IAM
- Konfigurasikan Lake Formation untuk kluster EMR yang diaktifkan Pusat Identitas IAM

Konfigurasikan Lake Formation untuk kluster EMR yang diaktifkan Pusat Identitas IAM

Anda dapat berintegrasi AWS Lake Formationdengan AWS IAM Identity Center kluster EMR yang diaktifkan.

Pertama, pastikan Anda memiliki instance Identity Center yang disiapkan di Region yang sama dengan cluster Anda. Untuk informasi selengkapnya, lihat Buat instance Pusat Identitas. Anda dapat menemukan ARN instance di konsol IAM Identity Center saat Anda melihat detail instance, atau menggunakan perintah berikut untuk melihat detail semua instance Anda dari CLI:

```
aws sso list-instances
```

Kemudian gunakan ARN dan ID AWS akun Anda dengan perintah berikut untuk mengonfigurasi Lake Formation agar kompatibel dengan IAM Identity Center:

```
aws lakeformation create-lake-formation-identity-center-configuration --cli-input-json
 file://create-lake-fromation-idc-config.json
json input:
{
    "CatalogId": "account-id/org-account-id",
    "InstanceArn": "identity-center-instance-arn"
}
```

Sekarang, hubungi `put-data-lake-settings` dan aktifkan `AllowFullTableExternalDataAccess` dengan Lake Formation:

```
aws lakeformation put-data-lake-settings --cli-input-json file://put-data-lake-
settings.json
json input:
{
    "DataLakeSettings": {
        "DataLakeAdmins": [
            {
                "DataLakePrincipalIdentifier": "admin-ARN"
            }
        ],
        "CreateDatabaseDefaultPermissions": [...],
        "CreateTableDefaultPermissions": [...],
        "AllowExternalDataFiltering": true,
        "AllowFullTableExternalDataAccess": true
    }
}
```

Terakhir, berikan izin tabel lengkap ke ARN identitas untuk pengguna yang mengakses cluster EMR. ARN berisi ID pengguna dari Pusat Identitas. Arahkan ke Pusat Identitas di konsol, pilih Pengguna, lalu pilih pengguna untuk melihat setelan informasi umum mereka.

Salin ID Pengguna dan tempel ke ARN berikut untuk: *user-id*

```
arn:aws:identitystore:::user/user-id
```

> **Note**
>
> Kueri pada kluster EMR hanya berfungsi jika identitas Pusat Identitas IAM memiliki akses tabel penuh pada tabel yang dilindungi Lake Formation. Jika identitas tidak memiliki akses tabel penuh, maka kueri akan gagal.

Gunakan perintah berikut untuk memberikan pengguna akses tabel penuh:

```
aws lakeformation grant-permissions --cli-input-json file://grantpermissions.json
json input:
{
    "Principal": {
        "DataLakePrincipalIdentifier": "arn:aws:identitystore:::user/user-id"
    },
    "Resource": {
        "Table": {
            "DatabaseName": "tip_db",
            "Name": "tip_table"
        }
    },
    "Permissions": [
        "ALL"
    ],
    "PermissionsWithGrantOption": [
        "ALL"
    ]
}
```

Bekerja dengan Hibah Akses S3 pada kluster EMR yang diaktifkan Pusat Identitas IAM

Anda dapat mengintegrasikan S3 Access Grants dengan kluster AWS IAM Identity Center EMR yang diaktifkan.

Gunakan S3 Access Grants untuk mengotorisasi akses ke kumpulan data Anda dari cluster yang menggunakan Identity Center. Buat hibah untuk menambah izin yang Anda tetapkan untuk pengguna IAM, grup, peran, atau untuk direktori perusahaan. Untuk informasi selengkapnya, lihat Menggunakan Hibah Akses S3 dengan Amazon EMR.

Topik

- Buat instance dan lokasi S3 Access Grants

- Buat hibah untuk identitas Pusat Identitas

Buat instance dan lokasi S3 Access Grants

Jika Anda belum memilikinya, buat instance S3 Access Grants di Wilayah AWS tempat Anda ingin meluncurkan cluster EMR Anda.

Gunakan AWS CLI perintah berikut untuk membuat instance baru bernama*MyInstance*:

```
aws s3control-access-grants create-access-grants-instance \
--account-id 12345678912 \
--identity-center-arn "identity-center-instance-arn" \
```

Kemudian, buat lokasi S3 Access Grants, ganti nilai merah dengan milik Anda sendiri:

```
aws s3control-access-grants create-access-grants-location \
--account-id 12345678912 \
--location-scope s3:// \
--iam-role-arn "access-grant-role-arn" \
--region aa-example-1
```

> **Note**
>
> Tentukan `iam-role-arn` parameter sebagai `accessGrantRole` ARN.

Buat hibah untuk identitas Pusat Identitas

Terakhir, buat hibah untuk identitas yang memiliki akses ke klaster Anda:

```
aws s3control-access-grants create-access-grant \
--account-id 12345678912 \
--access-grants-location-id "default" \
--access-grants-location-configuration S3SubPrefix="s3-bucket-prefix"
--permission READ \
--grantee GranteeType=DIRECTORY_USER,GranteeIdentifier="your-identity-center-user-id"
```

Contoh Keluaran:

```
{
```

```
"CreatedAt": "2023-09-21T23:47:24.870000+00:00",
"AccessGrantId": "1234-12345-1234-1234567",
"AccessGrantArn": "arn:aws:s3:aa-example-1-1:123456789012:access-grants/default/grant/
xxxx1234-1234-5678-1234-1234567890",
"Grantee": {
"GranteeType": "DIRECTORY_USER",
"GranteeIdentifier": "5678-56789-5678-567890"
},
"AccessGrantsLocationId": "default",
"AccessGrantsLocationConfiguration": {
"S3SubPrefix": "myprefix/*"
},
"Permission": "READ",
"GrantScope": "s3://myprefix/*"
}
```


Pertimbangan dan batasan untuk Amazon EMR dengan integrasi Pusat Identitas

Pertimbangkan poin-poin berikut saat Anda menggunakan IAM Identity Center dengan Amazon EMR:

- Propagasi identitas tepercaya melalui Identity Center didukung di Amazon EMR 6.15.0 dan lebih tinggi, dan hanya dengan Apache Spark.
- Untuk mengaktifkan kluster EMR dengan propagasi identitas tepercaya, Anda harus menggunakannya AWS CLI untuk membuat konfigurasi keamanan yang mengaktifkan propagasi identitas tepercaya, dan menggunakan konfigurasi keamanan tersebut saat meluncurkan klaster. Untuk informasi selengkapnya, lihat Membuat konfigurasi keamanan yang diaktifkan Pusat Identitas.
- Cluster EMR yang menggunakan propagasi identitas tepercaya hanya dapat memanggil layanan yang juga menggunakan propagasi identitas tepercaya.
- Hanya kontrol akses tingkat tabel berdasarkan AWS Lake Formation yang tersedia untuk kluster EMR yang menggunakan propagasi identitas tepercaya.
- Dengan kluster EMR yang menggunakan propagasi identitas tepercaya, operasi yang mendukung kontrol akses berdasarkan Lake Formation dengan Apache Spark meliputi,, dan. `SELECT` `ALTER TABLE` `DROP TABLE`
- Dengan kluster EMR yang menggunakan propagasi identitas tepercaya, kontrol akses berbasis Lake Formation yang tidak didukung dengan Apache Spark menyertakan pernyataan. `INSERT`
- Propagasi identitas tepercaya dengan Amazon EMR didukung sebagai berikut: Wilayah AWS

- `ap-east-1`— Asia Pasifik (Hong Kong) *
- `ap-northeast-1`— Asia Pasifik (Tokyo) *
- `ap-northeast-2`— Asia Pasifik (Seoul) *
- `ap-south-1`— Asia Pasifik (Mumbai) *
- `ap-southeast-1`— Asia Pasifik (Singapura)
- `ap-southeast-2`— Asia Pasifik (Sydney)
- `ca-central-1`— Kanada (Tengah)
- `eu-central-1`— Eropa (Frankfurt)
- `eu-north-1`— Eropa (Stockholm) *
- `eu-west-1`— Eropa (Irlandia)
- `eu-west-2`— Eropa (London)
- `eu-west-3`— Eropa (Paris) *
- `me-south-1`— Timur Tengah (Bahrain) *
- `sa-east-1`— Amerika Selatan (São Paulo) *
- `us-east-1`- AS Timur (Virginia N.)
- `us-east-2`— AS Timur (Ohio)
- `us-west-1`— AS Barat (California N.) *
- `us-west-2`— AS Barat (Oregon)

* Anda tidak dapat menggunakan Lake Formation dengan Amazon EMR dan propagasi identitas tepercaya di Wilayah ini.

Integrasi Amazon EMR dengan AWS Lake Formation

AWS Lake Formationadalah layanan terkelola yang membantu Anda menemukan, membuat katalog, membersihkan, dan mengamankan data di danau data Amazon Simple Storage Service (S3) Amazon Simple Storage Service (S3). Lake Formation menyediakan akses tingkat kolom berbutir halus ke database dan tabel di Katalog Data Glue. AWS Untuk informasi lebih lanjut, lihat Apa yang dimaksud dengan AWS Lake Formation?

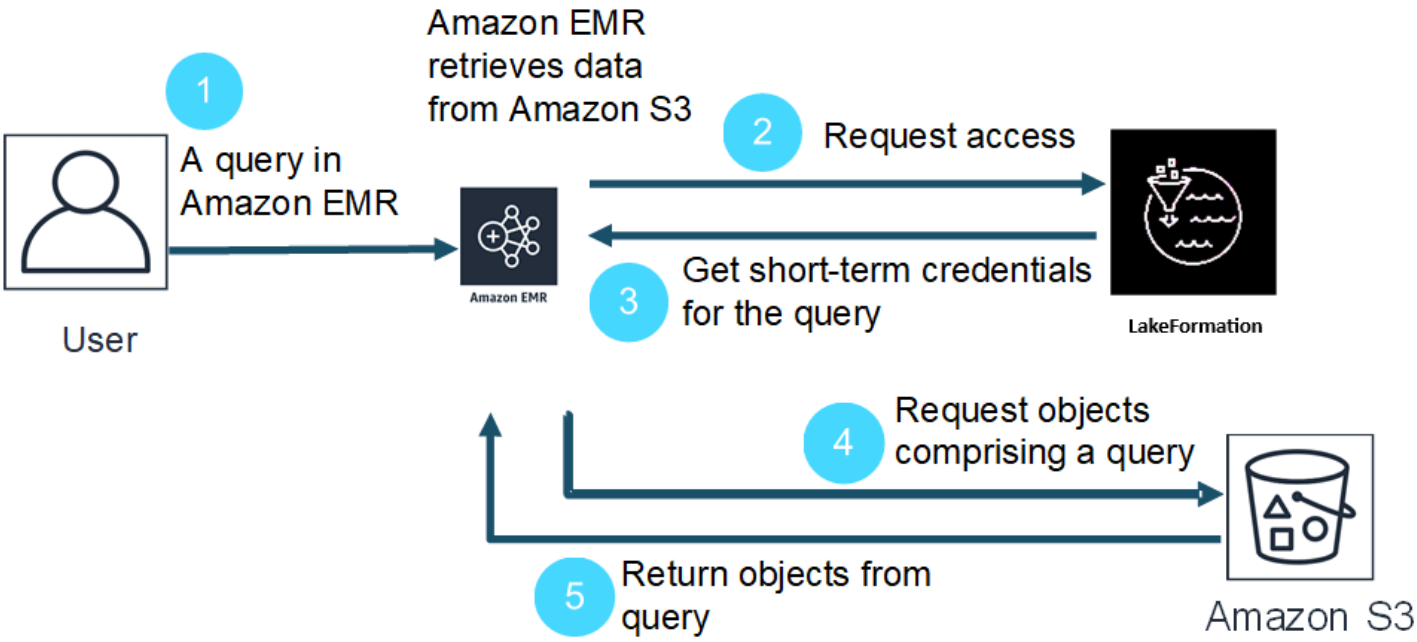
Dengan Amazon EMR rilis 6.7.0 dan yang lebih baru, Anda dapat menerapkan kontrol akses berbasis Lake Formation ke pekerjaan Spark, Hive, dan Presto yang Anda kirimkan ke kluster Amazon EMR. Untuk berintegrasi dengan Lake Formation, Anda harus membuat cluster EMR dengan peran

runtime. Peran runtime adalah peran AWS Identity and Access Management (IAM) yang Anda kaitkan dengan pekerjaan atau kueri EMR Amazon. Amazon EMR kemudian menggunakan peran ini untuk mengakses AWS sumber daya. Untuk informasi selengkapnya, lihat Peran runtime untuk langkah-langkah EMR Amazon.

Bagaimana Amazon EMR bekerja dengan Lake Formation

Setelah mengintegrasikan Amazon EMR dengan Lake Formation, Anda dapat menjalankan kueri ke klaster EMR Amazon dengan API atau dengan Studio. `Step` SageMaker Kemudian, Lake Formation menyediakan akses ke data melalui kredensyal sementara untuk Amazon EMR. Proses ini disebut credential vending. Untuk informasi lebih lanjut, lihat Apa yang dimaksud dengan AWS Lake Formation?

Berikut ini adalah ikhtisar tingkat tinggi tentang bagaimana Amazon EMR mendapatkan akses ke data yang dilindungi oleh kebijakan keamanan Lake Formation.



1. Seorang pengguna mengirimkan kueri EMR Amazon untuk data di Lake Formation.
2. Amazon EMR meminta kredensyal sementara dari Lake Formation untuk memberikan akses data pengguna.
3. Lake Formation mengembalikan kredensyal sementara.
4. Amazon EMR mengirimkan permintaan kueri untuk mengambil data dari Amazon S3.
5. Amazon EMR menerima data dari Amazon S3, memfilternya, dan mengembalikan hasil berdasarkan izin pengguna yang ditentukan pengguna di Lake Formation.

Untuk informasi selengkapnya tentang penambahan pengguna dan grup ke kebijakan Lake Formation, lihat Memberikan izin Katalog Data.

Prasyarat

Anda harus memenuhi persyaratan berikut sebelum mengintegrasikan Amazon EMR dan Lake Formation:

- Aktifkan otorisasi peran runtime di klaster EMR Amazon Anda.
- Gunakan AWS Glue Data Catalog sebagai toko metadata Anda.
- Menentukan dan mengelola izin di Lake Formation untuk mengakses basis data, tabel, dan kolom di AWS Katalog Data Glue. Untuk informasi lebih lanjut, lihat Apa yang dimaksud dengan AWS Lake Formation?

Topik

- Aktifkan Lake Formation dengan Amazon EMR
- Apache Hudi dan Lake Formation
- Gunung Es Apache dan Lake Formation
- Danau Delta dan Formasi Danau
- Pertimbangan untuk Amazon EMR dengan Lake Formation

Aktifkan Lake Formation dengan Amazon EMR

Dengan Amazon EMR 6.15.0 dan yang lebih tinggi, saat Anda menjalankan pekerjaan Spark di Amazon EMR di kluster EC2 yang mengakses data di Katalog Data AWS Glue, Anda dapat AWS Lake Formation menggunakannya untuk menerapkan izin tingkat tabel, baris, kolom, dan sel pada tabel berbasis Hudi, Iceberg, atau Delta Lake.

Di bagian ini, kami membahas cara membuat konfigurasi keamanan dan mengatur Lake Formation untuk bekerja dengan Amazon EMR. Kami juga membahas cara meluncurkan cluster dengan konfigurasi keamanan yang Anda buat untuk Lake Formation.

Langkah 1: Siapkan peran runtime untuk cluster EMR Anda

Untuk menggunakan peran runtime untuk klaster EMR Anda, Anda harus membuat konfigurasi keamanan. Dengan konfigurasi keamanan, Anda dapat menerapkan opsi keamanan, otorisasi, dan otentikasi yang konsisten di seluruh cluster Anda.

1. Buat file yang disebut `lf-runtime-roles-sec-cfg.json` dengan konfigurasi keamanan berikut.

```
{
    "AuthorizationConfiguration":{
        "IAMConfiguration":{
            "EnableApplicationScopedIAMRole":true,
            "ApplicationScopedIAMRoleConfiguration":{
                "PropagateSourceIdentity":true
            }
        },
        "LakeFormationConfiguration":{
            "AuthorizedSessionTagValue":"Amazon EMR"
        },
  "EncryptionConfiguration": {
            "EnableInTransitEncryption": true,
            "InTransitEncryptionConfiguration": {
            "TLSCertificateConfiguration": {<Certificate-configuration>}
        }
    }
}
```

2. Selanjutnya, untuk memastikan bahwa tag sesi dapat mengotorisasi Lake Formation, atur `LakeFormationConfiguration/AuthorizedSessionTagValue` properti ke`Amazon EMR`.
3. Gunakan perintah berikut untuk membuat konfigurasi keamanan Amazon EMR.

```
aws emr create-security-configuration \
--name 'iamconfig-with-iam-lf' \
--security-configuration file://lf-runtime-roles-sec-cfg.json
```

   Atau, Anda dapat menggunakan konsol EMR Amazon untuk membuat konfigurasi keamanan dengan pengaturan khusus.

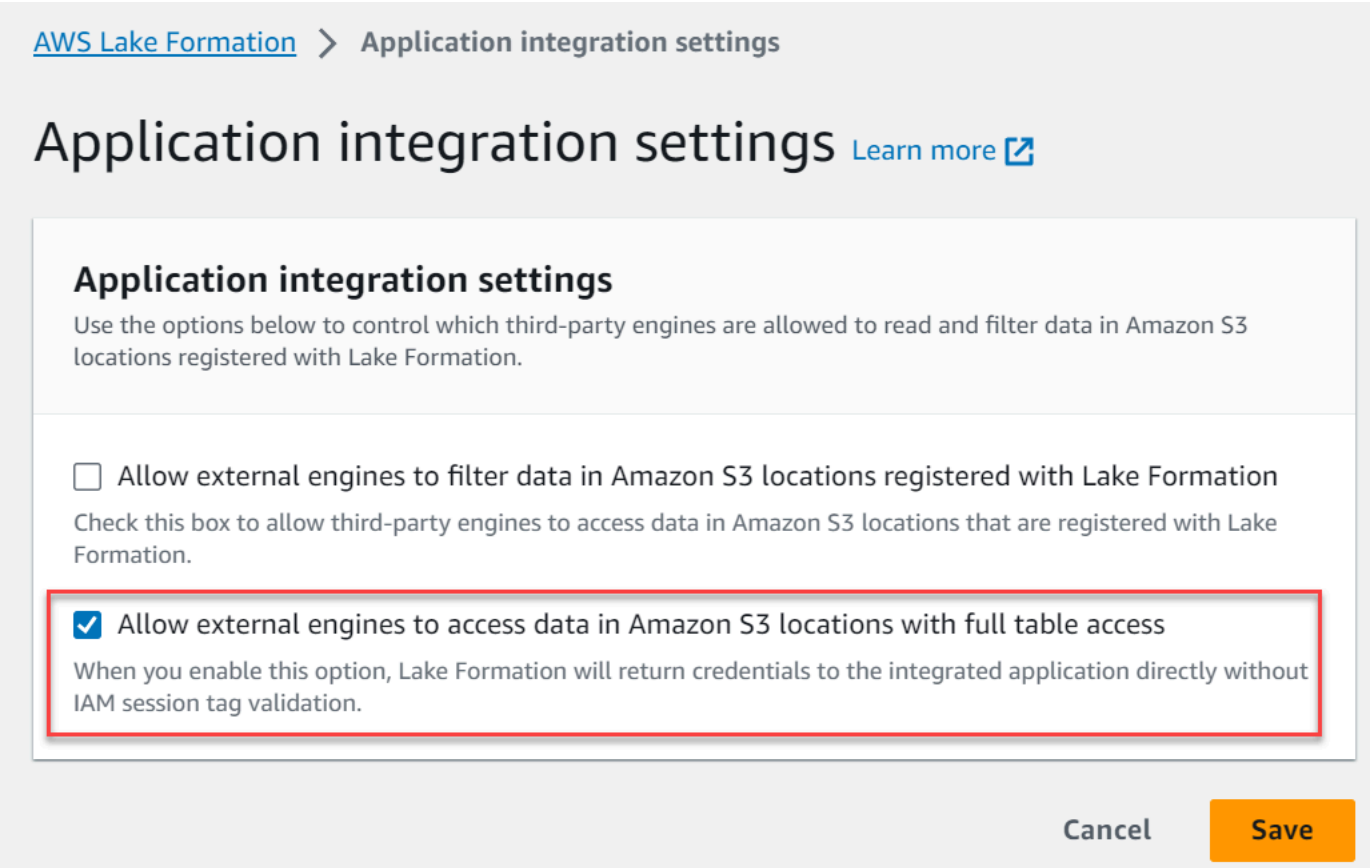
Langkah 2: Luncurkan cluster EMR Amazon

Sekarang Anda siap meluncurkan cluster EMR dengan konfigurasi keamanan yang Anda buat pada langkah sebelumnya. Untuk informasi selengkapnya tentang konfigurasi keamanan, lihat Menggunakan konfigurasi keamanan untuk mengatur keamanan klaster danPeran runtime untuk langkah-langkah EMR Amazon.

Langkah 3a: Siapkan izin tingkat tabel berbasis Lake Formation dengan peran runtime Amazon EMR

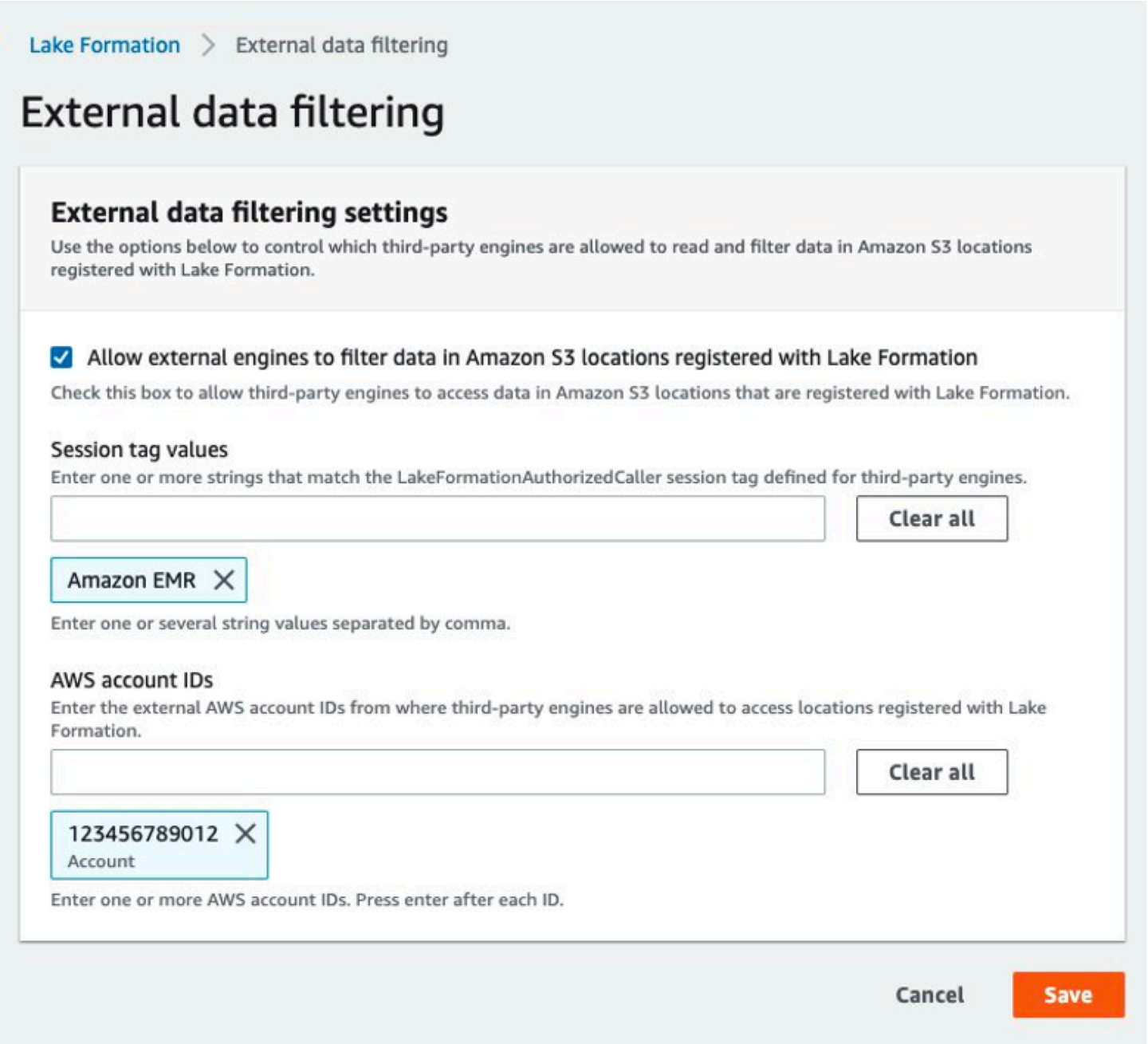
Jika Anda tidak memerlukan kontrol akses berbutir halus di kolom, baris, atau tingkat sel, Anda dapat mengatur izin tingkat tabel dengan Glue Data Catalog. Untuk mengaktifkan akses tingkat tabel, navigasikan ke AWS Lake Formation konsol dan pilih opsi Pengaturan integrasi aplikasi dari bagian Administrasi di bilah sisi. Kemudian, aktifkan opsi berikut dan pilih Simpan:

Izinkan mesin eksternal mengakses data di lokasi Amazon S3 dengan akses tabel penuh



Langkah 3b: Siapkan izin kolom, baris, atau tingkat sel berbasis Lake Formation dengan peran runtime Amazon EMR

Untuk menerapkan izin tingkat tabel dan kolom dengan Lake Formation, administrator danau data untuk Lake Formation harus menetapkan `Amazon EMR` sebagai nilai untuk konfigurasi tag sesi. `AuthorizedSessionTagValue` Lake Formation menggunakan tag sesi ini untuk mengotorisasi penelepon dan menyediakan akses ke danau data. Anda dapat mengatur tag sesi ini di bagian pemfilteran data eksternal pada konsol Lake Formation. Ganti *123456789012* dengan ID Anda sendiri. Akun AWS



Langkah 4: Konfigurasikan hibah AWS Glue dan Lake Formation untuk peran runtime Amazon EMR

Untuk melanjutkan penyiapan kontrol akses berbasis Lake Formation dengan peran runtime Amazon EMR, Anda harus mengonfigurasi hibah AWS Glue dan Lake Formation untuk peran runtime Amazon EMR. Untuk memungkinkan peran runtime IAM Anda berinteraksi dengan Lake Formation, beri mereka akses dengan `lakeformation:GetDataAccess` dan. `glue:Get*`

Izin Lake Formation mengontrol akses ke sumber daya Katalog Data AWS Glue, lokasi Amazon S3, dan data dasar di lokasi tersebut. Izin IAM mengontrol akses ke API dan sumber daya Lake Formation dan AWS Glue. Meskipun Anda mungkin memiliki izin Lake Formation untuk mengakses tabel di katalog data (SELECT), operasi Anda gagal jika Anda tidak memiliki izin IAM pada `glue:Get*` API. Untuk detail lebih lanjut tentang kontrol akses Lake Formation, lihat ikhtisar kontrol akses Lake Formation.

1. Buat `emr-runtime-roles-lake-formation-policy.json` file dengan konten berikut.

```
{
    "Version": "2012-10-17",
    "Statement": {
        "Sid": "LakeFormationManagedAccess",
        "Effect": "Allow",
        "Action": [
            "lakeformation:GetDataAccess",
            "glue:Get*",
            "glue:Create*",
            "glue:Update*"
        ],
        "Resource": "*"
    }
}
```

2. Buat kebijakan IAM terkait.

```
aws iam create-policy \
--policy-name emr-runtime-roles-lake-formation-policy \
--policy-document file://emr-runtime-roles-lake-formation-policy.json
```

3. Untuk menetapkan kebijakan ini ke peran runtime IAM Anda, ikuti langkah-langkah dalam Mengelola izin. AWS Lake Formation

Anda sekarang dapat menggunakan peran runtime dan Lake Formation untuk menerapkan izin tingkat tabel dan kolom. Anda juga dapat menggunakan identitas sumber untuk mengontrol tindakan dan memantau operasi denganAWS CloudTrail. Untuk selengkapnya, end-to-end lihat Memperkenalkan peran runtime untuk langkah-langkah EMR Amazon.

Apache Hudi dan Lake Formation

Amazon EMR merilis 6.15.0 dan yang lebih tinggi termasuk dukungan untuk kontrol akses berbutir halus berdasarkan Apache Hudi saat Anda membaca dan AWS Lake Formation menulis data dengan Spark SQL. Amazon EMR mendukung tabel, baris, kolom, dan kontrol akses tingkat sel dengan Apache Hudi. Dengan fitur ini, Anda dapat menjalankan kueri snapshot pada copy-on-write tabel untuk menanyakan snapshot terbaru dari tabel pada saat komit atau pemadatan tertentu.

Matriks dukungan berikut mencantumkan beberapa fitur inti Apache Hudi dengan Lake Formation:

	Salin di Tulis	Gabung saat Dibaca
Kueri snapshot - Spark SQL	✓	✓
Kueri yang dioptimalkan baca - Spark SQL	✓	✓
Kueri tambahan	✓	✓
Pertanyaan perjalanan waktu	✓	✓
Tabel metadata	✓	✓
Perintah DML **`INSERT`**	✓	✓
Perintah DDL		
Percikan kueri sumber data		
Sumber data Spark menulis		

Menanyakan tabel Hudi

Bagian ini menunjukkan bagaimana Anda dapat menjalankan kueri yang didukung yang dijelaskan di atas pada kluster yang diaktifkan Lake Formation. Tabel harus berupa tabel katalog terdaftar.

1. Untuk memulai shell Spark, gunakan perintah berikut.

```
spark-shell  --jars /usr/lib/hudi/hudi-spark-bundle.jar \
```

```
--conf 'spark.serializer=org.apache.spark.serializer.KryoSerializer'
```

```
spark-sql  --jars /usr/lib/hudi/hudi-spark-bundle.jar \
--conf 'spark.serializer=org.apache.spark.serializer.KryoSerializer'
```

2. Untuk menanyakan snapshot copy-on-write tabel terbaru, gunakan perintah berikut.

```
SELECT * FROM my_hudi_cow_table
```

```
spark.read.table("my_hudi_cow_table")
```

3. Untuk menanyakan data tabel terbaru yang dipadatkan, Anda dapat menanyakan `MOR` tabel yang dioptimalkan baca yang diakhiran dengan: `_ro`

```
SELECT * FROM my_hudi_mor_table_ro
```

```
spark.read.table("my_hudi_mor_table_ro")
```

> **Note**
>
> Kinerja pembacaan pada cluster Lake Formation mungkin lebih lambat karena optimasi yang tidak didukung. Fitur-fitur ini termasuk daftar file berdasarkan metadata Hudi, dan melewatkan data. Kami menyarankan Anda menguji kinerja aplikasi Anda untuk memastikan bahwa itu memenuhi persyaratan Anda.

Gunung Es Apache dan Lake Formation

Amazon EMR merilis 6.15.0 dan yang lebih tinggi termasuk dukungan untuk kontrol akses berbutir halus berdasarkan Apache Iceberg saat Anda membaca dan AWS Lake Formation menulis data dengan Spark SQL. Amazon EMR mendukung tabel, baris, kolom, dan kontrol akses tingkat sel dengan Apache Iceberg. Dengan fitur ini, Anda dapat menjalankan kueri snapshot pada copy-on-write tabel untuk menanyakan snapshot terbaru dari tabel pada saat komit atau pemadatan tertentu.

Jika Anda ingin menggunakan format Iceberg, atur konfigurasi berikut. Ganti *DB_LOCATION* dengan jalur Amazon S3 tempat tabel Iceberg Anda berada, dan ganti placeholder Region dan ID akun dengan nilai Anda sendiri.

```
spark-sql \
--conf
 spark.sql.extensions=org.apache.iceberg.spark.extensions.IcebergSparkSessionExtensions
 \
--conf spark.sql.catalog.my_catalog=org.apache.iceberg.spark.SparkCatalog \
--conf spark.sql.catalog.my_catalog.warehouse=DB_LOCATION \
--conf spark.sql.catalog.my_catalog.catalog-
impl=org.apache.iceberg.aws.glue.GlueCatalog \
--conf spark.sql.catalog.my_catalog.io-impl=org.apache.iceberg.aws.s3.S3FileIO \
--conf spark.sql.catalog.my_catalog.glue.lakeformation-enabled=true \
--conf spark.sql.catalog.my_catalog.client.region=aa-example-1 \
--conf
 spark.sql.catalog.my_catalog.client.factory=org.apache.iceberg.aws.lakeformation.LakeFormation
 \
--conf spark.sql.catalog.my_catalog.glue.account-id=ACCOUNT_ID
```

Anda juga harus berhati-hati untuk TIDAK melewati pengaturan peran asumsi berikut:

```
--conf spark.sql.catalog.my_catalog.client.assume-role.region
--conf spark.sql.catalog.my_catalog.client.assume-role.arn
--conf spark.sql.catalog.my_catalog.client.assume-
role.tags.LakeFormationAuthorizedCaller
```

Matriks dukungan berikut mencantumkan beberapa fitur inti Apache Iceberg dengan Lake Formation:

	Salin di Tulis	Gabung saat Dibaca
Kueri snapshot - Spark SQL	✓	✓
Kueri yang dioptimalkan baca - Spark SQL	✓	✓
Kueri tambahan	✓	✓
Pertanyaan perjalanan waktu	✓	✓
Tabel metadata	✓	✓
Perintah DML **`INSERT`**	✓	✓
Perintah DDL		

	Salin di Tulis	Gabung saat Dibaca
Percikan kueri sumber data		
Sumber data Spark menulis		

Danau Delta dan Formasi Danau

Amazon EMR merilis 6.15.0 dan yang lebih tinggi termasuk dukungan untuk kontrol akses berbutir halus berdasarkan Delta Lake saat Anda membaca dan AWS Lake Formation menulis data dengan Spark SQL. Amazon EMR mendukung tabel, baris, kolom, dan kontrol akses tingkat sel dengan Delta Lake. Dengan fitur ini, Anda dapat menjalankan kueri snapshot pada copy-on-write tabel untuk menanyakan snapshot terbaru dari tabel pada saat komit atau pemadatan tertentu.

Matriks dukungan berikut mencantumkan beberapa fitur inti Danau Delta dengan Lake Formation:

	Salin di Tulis	Gabung saat Dibaca
Kueri snapshot - Spark SQL	✓	✓
Kueri yang dioptimalkan baca - Spark SQL	✓	✓
Kueri tambahan	✓	✓
Pertanyaan perjalanan waktu	✓	✓
Tabel metadata	✓	✓
Perintah DML **`INSERT`**	✓	✓
Perintah DDL		
Percikan kueri sumber data		
Sumber data Spark menulis		

Pertimbangan untuk Amazon EMR dengan Lake Formation

Pertimbangkan hal berikut saat menggunakan Amazon EMR dengan. AWS Lake Formation

- Kontrol akses tingkat tabel tersedia di cluster dengan Amazon EMR rilis 6.13 dan lebih tinggi.
- Kontrol akses berbutir halus pada tingkat baris, kolom, dan sel tersedia di cluster dengan rilis Amazon EMR 6.15 dan lebih tinggi.
- Pengguna dengan akses ke tabel dapat mengakses semua properti tabel itu. Jika Anda memiliki kontrol akses berbasis Lake Formation pada tabel, tinjau tabel untuk memastikan bahwa properti tidak berisi data atau informasi sensitif apa pun.
- Cluster EMR Amazon dengan Lake Formation tidak mendukung fallback Spark ke HDFS saat Spark mengumpulkan statistik tabel. Ini biasanya membantu mengoptimalkan kinerja kueri.
- Operasi yang mendukung kontrol akses berdasarkan Lake Formation dengan tabel Apache Spark yang tidak diatur termasuk dan. `INSERT INTO` `INSERT OVERWRITE`
- Operasi yang mendukung kontrol akses berdasarkan Lake Formation dengan Apache Spark dan Apache Hive meliputi`SELECT`,,,, `DESCRIBE`, `SHOW DATABASE`, `SHOW TABLE` dan. `SHOW COLUMN` `SHOW PARTITION`
- Amazon EMR tidak mendukung kontrol akses ke operasi berbasis Lake Formation berikut:
  - Menulis ke tabel yang diatur
  - Filter data Lake Formation
  - Amazon EMR tidak mendukung. `CREATE TABLE` Amazon EMR 6.10.0 dan dukungan yang lebih tinggi. `ALTER TABLE`
  - Pernyataan DML selain `INSERT` perintah.
- Ada perbedaan kinerja antara kueri yang sama dengan dan tanpa kontrol akses berbasis Lake Formation.

Mengintegrasikan Amazon EMR dengan Apache Ranger

Dimulai dengan Amazon EMR 5.32.0, Anda dapat meluncurkan sebuah klaster yang secara alami terintegrasi dengan Apache Ranger. Apache Ranger adalah kerangka kerja sumber terbuka untuk mengaktifkan, memantau, dan mengelola keamanan data komprehensif di seluruh platform Hadoop. Untuk informasi selengkapnya, lihat Apache Ranger. Dengan integrasi alami, Anda dapat membawa Apache Ranger Anda sendiri untuk menegakkan kendali akses data lancar di Amazon EMR.

Bagian ini memberikan gambaran umum konseptual integrasi Amazon EMR dengan Apache Ranger. Hal ini juga mencakup prasyarat dan langkah-langkah yang diperlukan untuk meluncurkan klaster Amazon EMR yang terintegrasi dengan Apache Ranger.

Secara alami mengintegrasikan Amazon EMR dengan Apache Ranger memberikan manfaat kunci sebagai berikut:

- Kendali akses lancar ke basis data dan tabel Metastore Hive, yang mengizinkan Anda untuk menentukan kebijakan penyaringan data pada tingkat basis data, tabel, dan kolom untuk Apache Spark dan Apache Hive aplikasi. Penyaringan masking penyarignan dan data didukung dengan aplikasi Hive.
- Kemampuan untuk menggunakan kebijakan Hive yang ada langsung dengan Amazon EMR untuk aplikasi Hive.
- Kendali akses ke data Amazon S3 pada prefiks dan tingkat objek, yang mengizinkan Anda untuk menentukan kebijakan penyaringan data untuk akses ke data S3 menggunakan Sistem File EMR.
- Kemampuan untuk menggunakan CloudWatch Log untuk audit terpusat.
- Amazon EMR menginstal dan mengelola plugin Apache Ranger atas nama Anda.

Apache Ranger

Apache Ranger adalah kerangka kerja untuk mengaktifkan, memantau, dan mengelola keamanan data yang komprehensif di seluruh platform Hadoop.

Apache Ranger memiliki fitur-fitur berikut:

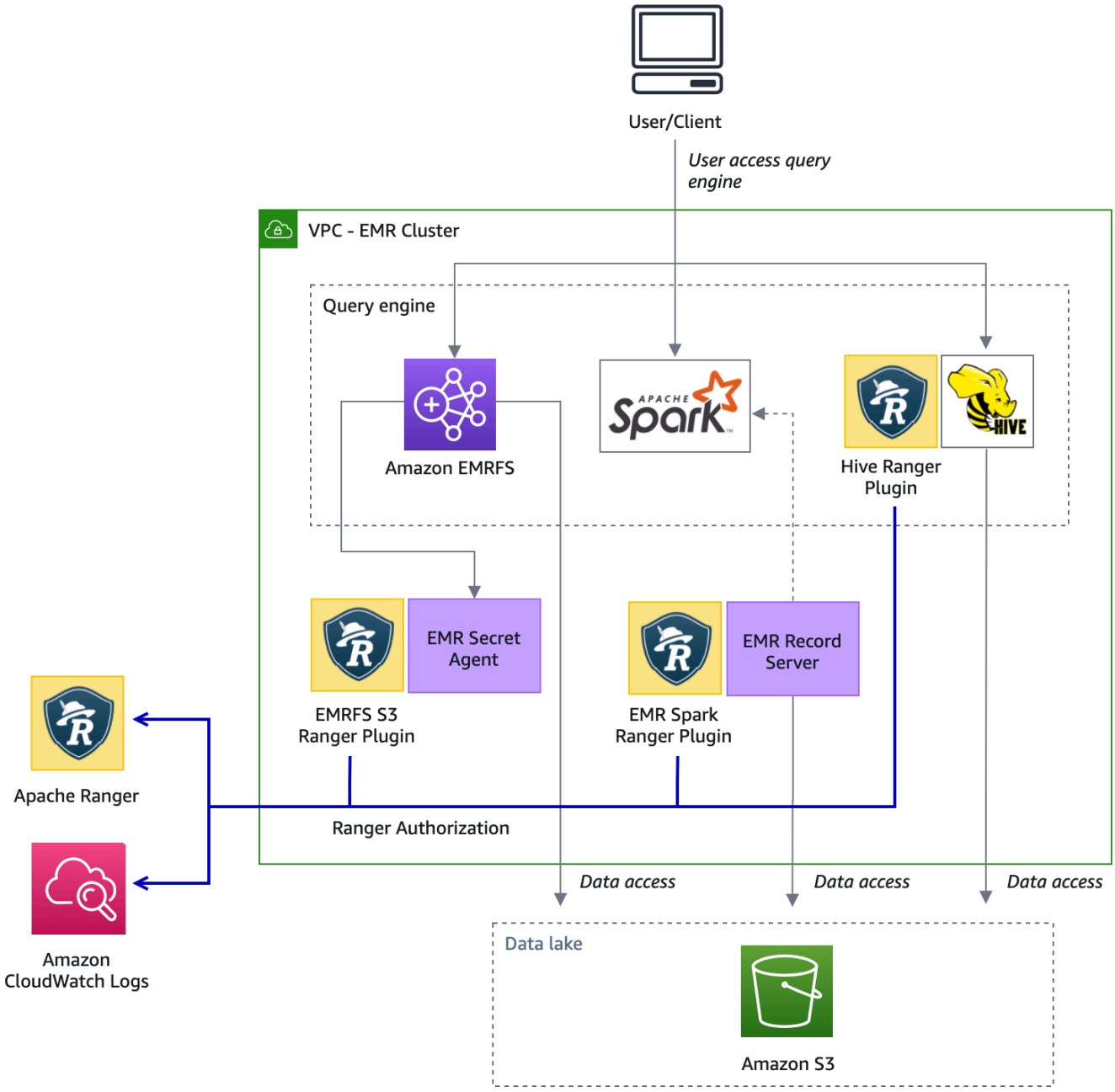
- Administrasi keamanan terpusat untuk mengelola semua tugas terkait keamanan di UI pusat atau menggunakan API REST.
- Otorisasi lancar untuk melakukan tindakan atau operasi tertentu dengan komponen Hadoop atau alat, dikelola melalui alat administrasi pusat.
- Sebuah metode otorisasi standar di semua komponen Hadoop.
- Support yang disempurnakan untuk berbagai metode otorisasi.
- Audit terpusat dari akses pengguna dan tindakan administratif (keamanan terkait) di semua komponen Hadoop.

Apache Ranger menggunakan dua komponen kunci untuk otorisasi:

- Server admin kebijakan Apache Ranger - Server ini mengizinkan Anda untuk menentukan kebijakan otorisasi untuk aplikasi Hadoop. Saat mengintegrasikan dengan Amazon EMR, Anda dapat menentukan dan menegakkan kebijakan untuk Apache Spark dan Hive untuk mengakses Hive Metastore, dan mengakses Sistem File EMR data Amazon S3 (EMRFS). Anda dapat membuat pengaturan baru atau menggunakan server admin kebijakan Apache Ranger yang ada untuk mengintegrasikan dengan Amazon EMR.
- Plugin Apache Ranger - Plugin ini memvalidasi akses pengguna terhadap kebijakan otorisasi yang didefinisikan di server admin kebijakan Apache Ranger. Amazon EMR menginstal dan mengonfigurasi plugin Apache Ranger secara otomatis untuk setiap aplikasi Hadoop yang dipilih di konfigurasi Apache Ranger.

Topik

Arsitektur integrasi Amazon EMR dengan Apache Ranger



Komponen Amazon EMR

Amazon EMR mengizinkan kendali akses lancar dengan Apache Ranger melalui komponen-komponen berikut. Lihat Diagram arsitektur untuk representasi visual dari komponen Amazon EMR ini dengan plugin Apache Ranger.

Agen rahasia – Agen rahasia secara aman menyimpan rahasia dan mendistribusikan rahasia ke komponen Amazon EMR atau aplikasi lain. Rahasia dapat mencakup kredensial pengguna sementara, kunci enkripsi, atau tiket Kerberos. Agen rahasia berjalan pada setiap simpul di klaster dan mencegat panggilan ke Layanan Metadata Instans. Untuk permintaan ke kredensial peran profil instans, Agen Rahasia menyediakan kredensialnya tergantung pada pengguna yang meminta dan sumber daya yang diminta setelah mengotorisasi permintaan dengan plugin Ranger S3 EMRFS. Agen rahasia berjalan sebagai pengguna *`emrsecretagent`*, dan menulis log ke direktori /emr/secretagent/log. Proses ini bergantung pada satu set aturan `iptables` tertentu untuk berfungsi. Penting untuk memastikan bahwa `iptables` tidak dinonaktifkan. Jika Anda menyesuaikan konfigurasi `iptables`, aturan tabel NAT harus dipertahankan dan dibiarkan tidak berubah.

Server catatan EMR – Server catatan menerima permintaan untuk mengakses data dari Spark. Kemudian mengotorisasi permintaan dengan meneruskan sumber daya yang diminta ke plugin Spark Ranger untuk Amazon EMR. Catatan server membaca data dari Amazon S3 dan mengembalikan data tingkat kolom bahwa pengguna diotorisasi untuk mengakses berdasarkan kebijakan Ranger. Server rekaman berjalan pada setiap node di cluster sebagai pengguna emr_record_server dan menulis log ke direktori /var/log/. emr-record-server

Support dan batasan aplikasi

Aplikasi-aplikasi yang didukung

Integrasi antara Amazon EMR dan Apache Ranger di mana EMR menginstal plugin Ranger saat ini mendukung aplikasi berikut:

- Apache Spark (Tersedia dengan EMR 5.32+ dan EMR 6.3+)
- Apache Hive (Tersedia dengan EMR 5.32+ dan EMR 6.3+)
- Akses S3 melalui EMRFS (Tersedia dengan EMR 5.32+ dan EMR 6.3+)

Aplikasi berikut dapat diinstal pada klaster EMR dan mungkin perlu dikonfigurasi untuk memenuhi kebutuhan keamanan Anda:

- Apache Hadoop (Tersedia dengan EMR 5.32+ dan EMR 6.3+ termasuk YARN dan HDFS)
- Apache Livy (Tersedia dengan EMR 5.32+ dan EMR 6.3+)
- Apache Zeppelin (Tersedia dengan EMR 5.32+ dan EMR 6.3+)
- Apache Hue (Tersedia dengan EMR 5.32+ dan EMR 6.3+)

- Ganglia (Tersedia dengan EMR 5.32+ dan EMR 6.3+)
- HCatalog (Tersedia dengan EMR 5.32+ dan EMR 6.3+)
- Mahout (Tersedia dengan EMR 5.32+ dan EMR 6.3+)
- MXNet (Tersedia dengan EMR 5.32+ dan EMR 6.3+)
- TensorFlow (Tersedia dengan EMR 5.32+ dan EMR 6.3+)
- Tez (Tersedia dengan EMR 5.32+ dan EMR 6.3+)
- Trino (Tersedia dengan EMR 6.7+)
- ZooKeeper (Tersedia dengan EMR 5.32+ dan EMR 6.3+)

Important

Aplikasi yang tercantum di atas adalah satu-satunya aplikasi yang saat ini didukung. Untuk memastikan keamanan klaster, Anda diperbolehkan untuk membuat klaster EMR dengan hanya aplikasi di daftar di atas ketika Apache Ranger diaktifkan.
Aplikasi lain saat ini tidak didukung. Untuk memastikan keamanan klaster Anda, mencoba untuk menginstal aplikasi lain akan menyebabkan penolakan klaster Anda.

Fitur yang didukung

Berikut fitur Amazon EMR dapat digunakan dengan Amazon EMR dan Apache Ranger:

- Enkripsi saat istirahat dan dalam transit
- Autentikasi Kerberos (diperlukan)
- Grup klaster, armada instans, dan instans Spot
- Konfigurasi ulang aplikasi pada klaster berjalan
- Server-side encryption (SSE) EMRFS

Note

Pengaturan enkripsi Amazon EMR mengatur SSE. Untuk informasi selengkapnya, lihat Opsi Enkripsi.

Batasan aplikasi

Ada beberapa keterbatasan yang perlu diingat ketika Anda mengintegrasikan Amazon EMR dan Apache Ranger:

- Anda tidak dapat menggunakan konsol untuk membuat konfigurasi keamanan yang menentukan AWS opsi integrasi Ranger di AWS GovCloud (US) Region. Konfigurasi keamanan dapat dilakukan dengan menggunakan CLI.
- Kerberos harus diinstal pada klaster Anda.
- UI Aplikasi (antarmuka pengguna) seperti UI Manajer Sumber Daya YARN, UI HDFS, dan NameNode UI Livy tidak disetel dengan otentikasi secara default.
- Izin HDFS default `umask` dikonfigurasi sehingga objek yang dibuat diatur ke `world wide readable` secara default.
- Amazon EMR tidak mendukung mode ketersediaan tinggi (beberapa primer) dengan Apache Ranger.
- Untuk batasan tambahan, lihat batasan untuk setiap aplikasi.

> **Note**
>
> Pengaturan enkripsi Amazon EMR mengatur SSE. Untuk informasi selengkapnya, lihat Opsi Enkripsi.

Batasan plugin

Setiap plugin memiliki batasan khusus. Untuk batasan plugin Apache Hive, lihat Batasan plugin Apache Hive. Untuk batasan plugin Apache Spark, lihat Batasan plugin Apache Spark. Untuk batasan plugin EMRFS S3, lihat batasan plugin EMRFS S3.

Atur Amazon EMR untuk Apache Ranger

Sebelum Anda menginstal Apache Ranger, tinjau informasi di bagian ini untuk memastikan bahwa Amazon EMR dikonfigurasi dengan benar.

Topik

- Atur server Admin Ranger

- IAM role untuk integrasi alami dengan Apache Ranger
- Buat konfigurasi keamanan EMR
- Menyimpan sertifikat TLS di AWS Secrets Manager
- Mulai klaster EMR
- Konfigurasikan Zeppelin untuk kluster EMR Amazon yang mendukung Apache Ranger
- Masalah yang diketahui

Atur server Admin Ranger

Untuk integrasi Amazon EMR, plugin aplikasi Apache Ranger harus berkomunikasi dengan server Admin menggunakan TLS/SSL.

Prasyarat: Pengaktifan SSL Server Admin Ranger

Apache Ranger di Amazon EMR membutuhkan komunikasi SSL dua arah antara plugin dan server Admin Ranger. Untuk memastikan bahwa plugin berkomunikasi dengan server Apache Ranger melalui SSL, aktifkan atribut berikut ranger-admin-site dalam.xml. pada server Admin Ranger.

```
<property>
    <name>ranger.service.https.attrib.ssl.enabled</name>
    <value>true</value>
</property>
```

Selain itu, konfigurasi berikut diperlukan.

```
<property>
    <name>ranger.https.attrib.keystore.file</name>
    <value>_<PATH_TO_KEYSTORE>_</value>
</property>

<property>
    <name>ranger.service.https.attrib.keystore.file</name>
    <value>_<PATH_TO_KEYSTORE>_</value>
</property>

<property>
    <name>ranger.service.https.attrib.keystore.pass</name>
    <value>_<KEYSTORE_PASSWORD>_</value>
```

```
</property>

<property>
    <name>ranger.service.https.attrib.keystore.keyalias</name>
    <value><PRIVATE_CERTIFICATE_KEY_ALIAS></value>
</property>

<property>
    <name>ranger.service.https.attrib.clientAuth</name>
    <value>want</value>
</property>

<property>
    <name>ranger.service.https.port</name>
    <value>6182</value>
</property>
```

Sertifikat TLS

Integrasi Apache Ranger dengan Amazon EMR mengharuskan lalu lintas dari simpul Amazon EMR ke server Admin Ranger dienkripsi menggunakan TLS, dan bahwa Plugin Ranger mengautentikasi ke server Apache Ranger menggunakan dua arah autentikasi TLS. Layanan Amazon EMR membutuhkan sertifikat publik dari server Admin Ranger Anda (ditentukan di contoh sebelumnya) dan sertifikat privat.

Sertifikat plugin Apache Ranger

Sertifikat TLS publik plugin Apache Ranger harus dapat diakses ke server Admin Apache Ranger untuk memvalidasi ketika plugin connect. Ada tiga metode berbeda untuk melakukan hal ini.

Metode 1: Konfigurasikan truststore di server Admin Apache Ranger

Isi konfigurasi berikut ranger-admin-site di.xml. untuk mengkonfigurasi truststore.

```
<property>
    <name>ranger.truststore.file</name>
    <value><LOCATION TO TRUSTSTORE></value>
</property>

<property>
    <name>ranger.truststore.password</name>
    <value><PASSWORD FOR TRUSTSTORE></value>
```

```
</property>
```

Metode 2: Muat sertifikat ke Java cacerts truststore

Jika server Admin Ranger Anda tidak menentukan truststore di pilihan JVM, maka Anda dapat menempatkan sertifikat publik plugin di penyimpanan cacerts default.

Metode 3: Buat truststore dan tentukan sebagai bagian dari JVM Options

Di `{RANGER_HOME_DIRECTORY}/ews/ranger-admin-services.sh`, modifikasi `JAVA_OPTS` termasuk `"-Djavax.net.ssl.trustStore=<TRUSTSTORE_LOCATION>"` dan `"-Djavax.net.ssl.trustStorePassword=<TRUSTSTORE_PASSWORD>"`. Misalnya, menambahkan baris berikut setelah JAVA_OPTS yang ada.

```
JAVA_OPTS=" ${JAVA_OPTS} -Djavax.net.ssl.trustStore=${RANGER_HOME}/truststore/
truststore.jck -Djavax.net.ssl.trustStorePassword=changeit"
```

> **Note**
>
> Spesifikasi ini dapat mengekspos kata sandi truststore jika setiap pengguna dapat masuk ke server Admin Apache Ranger dan melihat proses yang berjalan, seperti ketika menggunakan perintah `ps`.

Menggunakan Sertifikat Self-Signed

Sertifikat bertandatangan sendiri tidak direkomendasikan sebagai sertifikat. Sertifikat yang bertandatangan sendiri mungkin tidak dicabut, dan sertifikat yang bertandatangan sendiri mungkin tidak sesuai dengan persyaratan keamanan internal.

Penginstalan definisi layanan

Definisi layanan yang digunakan oleh server Admin Ranger untuk mengcitrakan atribut kebijakan untuk aplikasi. Kebijakan tersebut kemudian disimpan di repositori kebijakan bagi klien untuk mengunduh.

Untuk dapat mengonfigurasi definisi layanan, panggilan REST harus dibuat ke server Admin Ranger. Lihat Apache Ranger PublicAPIsv2 untuk API yang diperlukan di bagian berikut.

Menginstal Definisi Layanan Apache Spark

Untuk menginstal Definisi Layanan Apache Spark, lihat Plugin Apache Spark.

Menginstal Definisi Layanan EMRFS

Untuk menginstal definisi layanan S3 untuk Amazon EMR, lihat Plugin S3 EMRFS.

Menggunakan Definisi Layanan Hive

Apache Hive dapat menggunakan definisi layanan Ranger yang dikirim dengan Apache Ranger 2.0 dan versi terbaru. Untuk informasi selengkapnya, lihat Plugin Apache Hive.

Aturan lalu lintas jaringan

Ketika Apache Ranger terintegrasi dengan klaster EMR Anda, klaster perlu berkomunikasi dengan server tambahan dan AWS.

Semua simpul Amazon EMR, termasuk simpul inti dan simpul tugas, harus mampu berkomunikasi dengan server Admin Apache Ranger untuk mengunduh kebijakan. Jika Admin Apache Ranger berjalan di Amazon EC2, Anda perlu memperbarui grup keamanan untuk dapat mengambil lalu lintas dari klaster EMR.

Selain berkomunikasi dengan Server Admin Ranger, semua simpul perlu dapat berkomunikasi dengan yang layanan AWS berikut:

- Amazon S3
- AWS KMS (jika menggunakan SSE-KMS EMRFS)
- Amazon CloudWatch
- AWS STS

Jika Anda berencana untuk menjalankan klaster EMR Anda di subnet privat, konfigurasi VPC untuk dapat berkomunikasi dengan layanan ini menggunakan AWS PrivateLink dan VPC endpoint di Panduan Pengguna Amazon VPC atau menggunakan instans terjemahan alamat instans (NAT) di Panduan Pengguna Amazon VPC.

IAM role untuk integrasi alami dengan Apache Ranger

Integrasi antara Amazon EMR dan Apache Ranger bergantung pada tiga peran kunci yang harus Anda buat sebelum Anda meluncurkan klaster Anda:

- Profil instans Amazon EC2 kustom untuk Amazon EMR

- IAM role untuk Mesin Apache Ranger
- Sebuah IAM role untuk layanan AWS lain

Bagian ini memberikan gambaran umum peran ini dan kebijakan yang perlu Anda sertakan untuk setiap IAM role. Untuk informasi tentang membuat peran, lihat Atur server Admin Ranger.

Profil instans EC2

Amazon EMR menggunakan peran layanan IAM untuk melakukan tindakan atas nama Anda untuk menyediakan dan mengelola klaster. Peran layanan untuk instans EC2 klaster, juga disebut profil instans EC2 untuk Amazon EMR, adalah tipe khusus dari peran layanan yang ditugaskan untuk setiap instans EC2 di sebuah klaster pada peluncuran.

Untuk menentukan izin interaksi klaster EMR dengan data Amazon S3 dan metastore Hive yang dilindungi oleh Apache Ranger dan layanan AWS lainnya, tentukan profil instans EC2 kustom yang akan digunakan, bukan saat Anda meluncurkan klaster. `EMR_EC2_DefaultRole`

Untuk informasi lebih lanjut, lihat Peran layanan untuk instans EC2 klaster (profil instans EC2) dan Kustom IAM role.

Anda perlu menambahkan pernyataan berikut ke Profil Instans EC2 default untuk Amazon EMR untuk dapat memberi tanda pada sesi dan mengakses AWS Secrets Manager yang menyimpan sertifikat TLS.

```
    {
      "Sid": "AllowAssumeOfRolesAndTagging",
      "Effect": "Allow",
      "Action": ["sts:TagSession", "sts:AssumeRole"],
      "Resource": [
        "arn:aws:iam::<AWS_ACCOUNT_ID>:role/<RANGER_ENGINE-
PLUGIN_DATA_ACCESS_ROLE_NAME>",
        "arn:aws:iam::<AWS_ACCOUNT_ID>:role/<RANGER_USER_ACCESS_ROLE_NAME>"
      ]
    },
    {
        "Sid": "AllowSecretsRetrieval",
        "Effect": "Allow",
        "Action": "secretsmanager:GetSecretValue",
        "Resource": [

 "arn:aws:secretsmanager:<REGION>:<AWS_ACCOUNT_ID>:secret:<PLUGIN_TLS_SECRET_NAME>*",
```

```
"arn:aws:secretsmanager:<REGION>:<AWS_ACCOUNT_ID>:secret:<ADMIN_RANGER_SERVER_TLS_SECRET_NAME>
      ]
  }
```

> **Note**
>
> Untuk izin Secrets Manager, jangan lupa wildcard ("*") di akhir nama rahasia atau permintaan Anda akan gagal. Wildcard adalah untuk versi rahasia.

> **Note**
>
> Membatasi cakupan kebijakan AWS Secrets Manager hanya untuk sertifikat yang diperlukan untuk penyediaan.

IAM role untuk Apache Ranger

Peran ini memberikan kredensyal untuk mesin eksekusi tepercaya, seperti Apache Hive dan Amazon EMR Record Server untuk mengakses data Amazon S3. Gunakan hanya peran ini untuk mengakses data Amazon S3, termasuk kunci KMS, jika Anda menggunakan SSE-KMS S3.

Peran ini harus dibuat dengan kebijakan minimum yang dinyatakan di contoh berikut.

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

 "arn:aws:logs:<REGION>:<AWS_ACCOUNT_ID>:<CLOUDWATCH_LOG_GROUP_NAME_IN_SECURITY_CONFIGURATION>:
      ]
    },
```

```
    {
      "Sid": "BucketPermissionsInS3Buckets",
      "Action": [
        "s3:CreateBucket",
        "s3:DeleteBucket",
        "s3:ListAllMyBuckets",
        "s3:ListBucket"
      ],
      "Effect": "Allow",
      "Resource": [
 *"arn:aws:s3:::bucket1",
        "arn:aws:s3:::bucket2"*
        ]
    },
    {
      "Sid": "ObjectPermissionsInS3Objects",
      "Action": [
        "s3:GetObject",
        "s3:DeleteObject",
        "s3:PutObject"
      ],
      "Effect": "Allow",
      "Resource": [
 *"arn:aws:s3:::bucket1/*",
          "arn:aws:s3:::bucket2/*"
*        ]
    }
  ]
}
```

Important

Tanda bintang “*” di akhir Sumber Daya CloudWatch Log harus disertakan untuk memberikan izin menulis ke aliran log.

Note

Jika Anda menggunakan tampilan konsistensi EMRFS atau enkripsi S3-SSE, Anda perlu menambahkan izin ke tabel DynamoDB dan kunci KMS sehingga mesin eksekusi dapat berinteraksi dengan mesin tersebut.

IAM role untuk Apache Ranger diasumsikan oleh Peran Profil Instans EC2. Gunakan contoh berikut untuk membuat kebijakan kepercayaan yang mengizinkan IAM role untuk Apache Ranger diasumsikan oleh peran profil instans EC2.

```
   {
     "Sid": "",
     "Effect": "Allow",
     "Principal": {
       "AWS": "arn:aws:iam::<AWS_ACCOUNT_ID>:role/<EC2 INSTANCE PROFILE ROLE NAME eg.
EMR_EC2_DefaultRole>"
     },
     "Action": ["sts:AssumeRole", "sts:TagSession"]
   }
```

IAM role untuk layanan AWS lain

Peran ini menyediakan eksekusi mesin dengan kredensial untuk berinteraksi dengan layanan AWS bagi pengguna yang tidak percaya, jika diperlukan. Jangan gunakan IAM role ini untuk mengizinkan akses ke data Amazon S3, kecuali itu adalah data yang harus dapat diakses oleh semua pengguna.

Peran ini akan diasumsikan oleh Peran Profil Instans EC2. Gunakan contoh berikut untuk membuat kebijakan kepercayaan yang mengizinkan IAM role untuk Apache Ranger diasumsikan oleh peran profil instans EC2.

```
   {
     "Sid": "",
     "Effect": "Allow",
     "Principal": {
       "AWS": "arn:aws:iam::<AWS_ACCOUNT_ID>:role/<EC2 INSTANCE PROFILE ROLE NAME eg.
EMR_EC2_DefaultRole>"
     },
     "Action": ["sts:AssumeRole", "sts:TagSession"]
   }
```

Memvalidasi izin Anda

Lihat Penyelesaian masalah Apache Ranger untuk petunjuk tentang memvalidasi izin.

Buat konfigurasi keamanan EMR

Membuat Konfigurasi Keamanan EMR Amazon untuk Apache Ranger

Sebelum Anda meluncurkan kluster EMR Amazon yang terintegrasi dengan Apache Ranger, buat konfigurasi keamanan.

Console

Untuk membuat konfigurasi keamanan yang menentukan AWS opsi integrasi Ranger

1. Di konsol Amazon EMR, pilih Konfigurasi keamanan, kemudian Buat.
2. Ketik Nama untuk konfigurasi keamanan. Anda menggunakan nama ini untuk menentukan konfigurasi keamanan ketika Anda membuat sebuah klaster.
3. Di bawah AWS Integrasi Ranger, memilih Aktifkan kendali akses lancar yang dikelola oleh Apache Ranger.
4. Pilih IAM role untuk Apache Ranger untuk diterapkan. Untuk informasi selengkapnya, lihat IAM role untuk integrasi alami dengan Apache Ranger.
5. Pilih IAM role Anda untuk layanan AWS lain untuk diterapkan.
6. Konfigurasikan plugin untuk connect ke server Admin Ranger dengan memasukkan ARN Secrets Manager untuk server Admin dan alamat.
7. Pilih aplikasi untuk mengkonfigurasi plugin Ranger. Isi ARN Secrets Manager yang berisi sertifikat TLS privat untuk plugin.

   Jika Anda tidak mengonfigurasi Apache Spark atau Apache Hive, dan mereka dipilih sebagai aplikasi untuk klaster Anda, permintaan gagal.
8. Mengatur opsi konfigurasi keamanan lain yang sesuai dan memilih Buat. Anda harus mengaktifkan autentikasi Kerberos menggunakan klaster khusus atau eksternal KDC.

> **Note**
>
> Anda tidak dapat menggunakan konsol untuk membuat konfigurasi keamanan yang menentukan AWS opsi integrasi Ranger di AWS GovCloud (US) Region. Konfigurasi keamanan dapat dilakukan dengan menggunakan CLI.

CLI

Untuk membuat konfigurasi keamanan untuk integrasi Apache Ranger

1. Ganti *<ACCOUNT ID>* dengan ID akun AWS Anda.

2. Ganti *<REGION>* dengan Wilayah tempat sumber daya berada.
3. Tentukan nilai untuk `TicketLifetimeInHours` dalam menentukan periode untuk tiket Kerberos valid yang dikeluarkan oleh KDC.
4. Tentukan alamat pelayan Admin Ranger untuk `AdminServerURL`.

```
{
    "AuthenticationConfiguration": {
        "KerberosConfiguration": {
            "Provider": "ClusterDedicatedKdc",
            "ClusterDedicatedKdcConfiguration": {
                "TicketLifetimeInHours": 24
            }
        }
    },
    "AuthorizationConfiguration":{
      "RangerConfiguration":{
         "AdminServerURL":"https://_<RANGER ADMIN SERVER IP>_:6182",
         "RoleForRangerPluginsARN":"arn:aws:iam::_<ACCOUNT ID>_:role/_<RANGER PLUGIN
 DATA ACCESS ROLE NAME>_",
         "RoleForOtherAWSServicesARN":"arn:aws:iam::_<ACCOUNT ID>_:role/_<USER
 ACCESS ROLE NAME>_",
         "AdminServerSecretARN":"arn:aws:secretsmanager:_<REGION>_:_<ACCOUNT
 ID>_:secret:_<SECRET NAME THAT PROVIDES ADMIN SERVERS PUBLIC TLS CERTIFICATE
 WITHOUT VERSION>_",
         "RangerPluginConfigurations":[
            {
               "App":"Spark",
               "ClientSecretARN":"arn:aws:secretsmanager:_<REGION>_:_<ACCOUNT
 ID>_:secret:_<SECRET NAME THAT PROVIDES SPARK PLUGIN PRIVATE TLS CERTIFICATE
 WITHOUT VERSION>_",
               "PolicyRepositoryName":"<SPARK SERVICE NAME eg. amazon-emr-spark>"
            },
            {
               "App":"Hive",
               "ClientSecretARN":"arn:aws:secretsmanager:_<REGION>_:_<ACCOUNT
 ID>_:secret:_<SECRET NAME THAT PROVIDES Hive PLUGIN PRIVATE TLS CERTIFICATE WITHOUT
 VERSION>_",
               "PolicyRepositoryName":"<HIVE SERVICE NAME eg. Hivedev>"
            },
            {
               "App":"EMRFS-S3",
```

```
                "ClientSecretARN":"arn:aws:secretsmanager:_<REGION>_:_<ACCOUNT
 ID>_:secret:_<SECRET NAME THAT PROVIDES EMRFS S3 PLUGIN PRIVATE TLS CERTIFICATE
 WITHOUT VERSION>_",
                "PolicyRepositoryName":"<EMRFS S3 SERVICE NAME eg amazon-emr-emrfs>"
            },
        {
                "App":"Trino",
                "ClientSecretARN":"arn:aws:secretsmanager:_<REGION>_:_<ACCOUNT
 ID>_:secret:_<SECRET NAME THAT PROVIDES TRINO PLUGIN PRIVATE TLS CERTIFICATE
 WITHOUT VERSION>_",
                "PolicyRepositoryName":"<TRINO SERVICE NAME eg amazon-emr-trino>"
            }
          ],
          "AuditConfiguration":{
             "Destinations":{
                "AmazonCloudWatchLogs":{
                   "CloudWatchLogGroup":"arn:aws:logs:<REGION>:_<ACCOUNT ID>_:log-
group:_<LOG GROUP NAME FOR AUDIT EVENTS>_"
                }
             }
          }
       }
    }
}
```

Itu PolicyRespositoryNames adalah nama layanan yang ditentukan dalam Admin Apache Ranger Anda.

Buat konfigurasi keamanan Amazon EMR dengan perintah berikut. Ganti konfigurasi keamanan dengan nama pilihan Anda. Memilih konfigurasi ini dengan nama ketika Anda membuat klaster Anda.

```
aws emr create-security-configuration \
--security-configuration file://./security-configuration.json \
--name security-configuration
```

Konfigurasikan Fitur Keamanan Tambahan

Untuk mengintegrasikan Amazon EMR dengan Apache Ranger, Anda juga harus mengonfigurasi fitur keamanan EMR berikut:

- Mengaktifkan autentikasi Kerberos menggunakan klaster khusus atau eksternal KDC. Untuk instruksi, lihat Gunakan Kerberos untuk otentikasi dengan Amazon EMR.
- (Opsional) Aktifkan enkripsi dalam transit atau saat istirahat. Untuk informasi selengkapnya, lihat Opsi enkripsi.

Untuk informasi selengkapnya, lihat Keamanan di Amazon EMR.

Menyimpan sertifikat TLS di AWS Secrets Manager

Plugin Ranger diinstal pada klaster Amazon EMR dan server Admin Ranger harus berkomunikasi melalui TLS untuk memastikan bahwa data kebijakan dan informasi lain yang dikirim tidak dapat dibaca jika mereka dicegat. EMR juga memberi mandat bahwa plugin mengautentikasi ke server Admin Ranger dengan menyediakan sertifikat TLS sendiri dan melakukan autentikasi TLS dua arah. Penataan ini memerlukan empat sertifikat yang akan dibuat: dua instal sertifikat TLS privat dan publik. Untuk petunjuk tentang menginstal sertifikat ke server Admin Ranger Anda, lihat Atur server Admin Ranger. Untuk menyelesaikan penataan, plugin Ranger diinstal pada klaster EMR yang memerlukan dua sertifikat: sertifikat TLS publik dari server admin Anda, dan sertifikat privat yang akan digunakan plugin untuk mengautentikasi terhadap server Admin Ranger. Untuk memberikan sertifikat TLS ini, mereka harus berada di AWS Secrets Manager dan disediakan di konfigurasi keamanan EMR.

> **Note**
>
> Sangat direkomendasikan, tetapi tidak diperlukan, untuk membuat instalan sertifikat untuk setiap aplikasi Anda untuk membatasi dampak jika salah satu sertifikat plugin menjadi terganggu.

> **Note**
>
> Anda harus melacak dan memutar sertifikat sebelum tanggal kedaluwarsa mereka.

Format Sertifikat

Mengimpor sertifikat ke AWS Secrets Manager adalah sama terlepas dari apakah itu adalah sertifikat plugin privat atau sertifikat admin Ranger publik. Sebelum mengimpor sertifikat TLS, sertifikat harus dalam format PEM 509x.

Contoh sertifikat publik dalam format:

```
-----BEGIN CERTIFICATE-----
...Certificate Body...
-----END CERTIFICATE-----
```

Contoh sertifikat privat dalam format:

```
-----BEGIN PRIVATE KEY-----
...Private Certificate Body...
-----END PRIVATE KEY-----
-----BEGIN CERTIFICATE-----
...Trust Certificate Body...
-----END CERTIFICATE-----
```

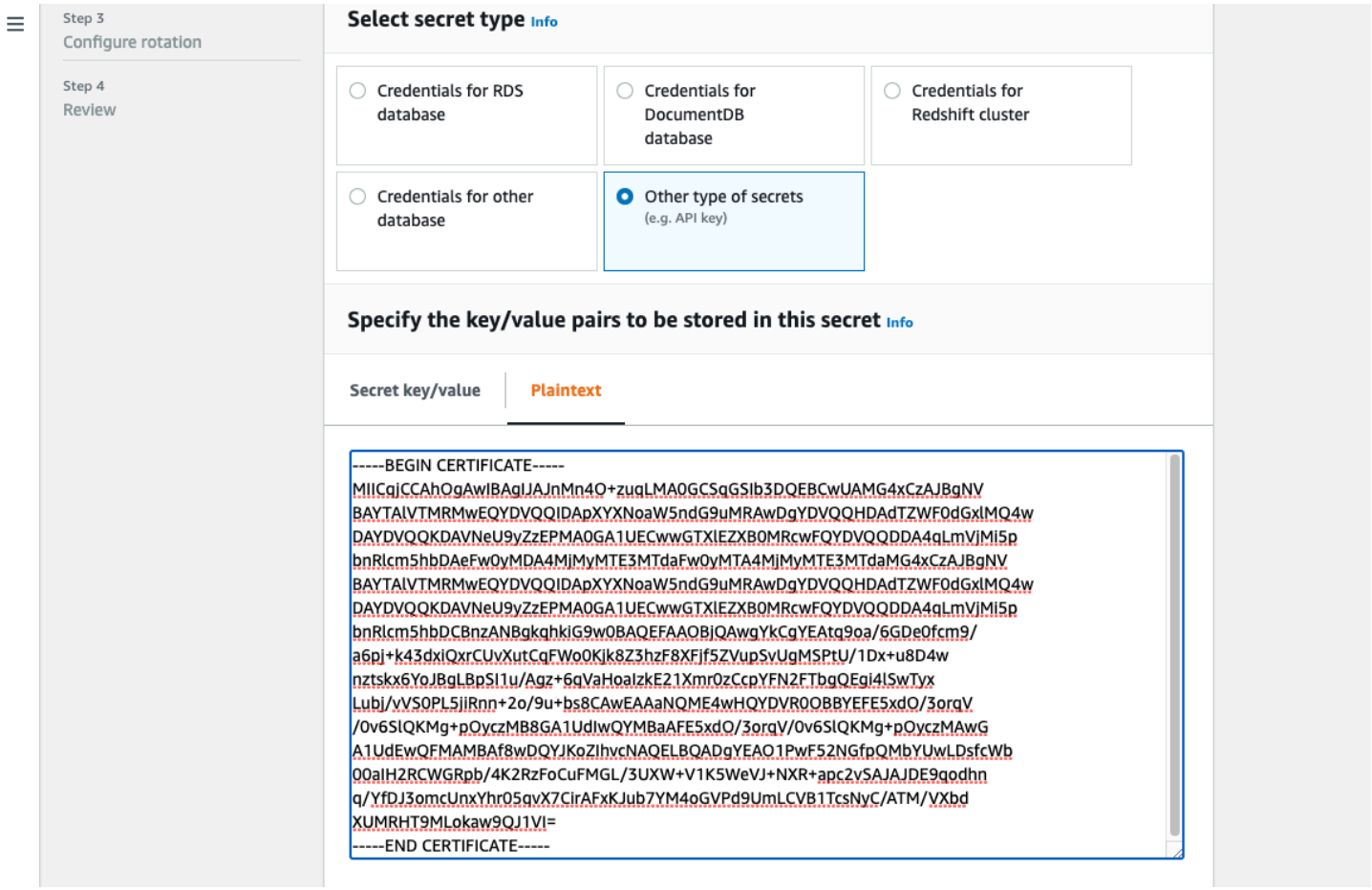
Sertifikat privat juga harus berisi sertifikat kepercayaan juga.

Anda dapat memvalidasi sertifikat dalam format yang benar dengan menjalankan perintah berikut:

```
openssl x509 -in <PEM FILE> -text
```

Mengimpor sertifikat ke AWS Secrets Manager

Saat membuat Secret Anda di Secrets Manager, pilih Jenis rahasia lain di bawah tipe rahasia dan tempel sertifikat yang dikodekan PEM Anda di bidang Plaintext.



Mulai klaster EMR

Sebelum meluncurkan klaster EMR Amazon dengan Apache Ranger, pastikan setiap komponen memenuhi persyaratan versi minimum berikut:

- Amazon EMR 5.32.0 atau yang lebih baru, atau 6.3.0 atau yang lebih baru. Kami menyarankan Anda menggunakan versi rilis Amazon EMR terbaru.
- Server Admin Apache Ranger 2.x.

Selesaikan langkah-langkah berikut:

- Instal Apache Ranger jika belum. Untuk informasi selengkapnya, lihat instalasi Apache Ranger 0.5.0.
- Pastikan ada konektivitas jaringan antara klaster Amazon EMR dan server Admin Apache Ranger. Lihat Atur server Admin Ranger
- Buat IAM role yang diperlukan. Lihat IAM role untuk integrasi alami dengan Apache Ranger.

- Buat konfigurasi keamanan EMR untuk instalasi Apache Ranger. Untuk informasi lebih lanjut, lihat Buat konfigurasi keamanan EMR.

Konfigurasikan Zeppelin untuk kluster EMR Amazon yang mendukung Apache Ranger

Topiknya mencakup cara mengonfigurasi Apache Zeppelin untuk kluster EMR Amazon yang mendukung Apache Ranger sehingga Anda dapat menggunakan Zeppelin sebagai buku catatan untuk eksplorasi data interaktif. Zeppelin disertakan dalam versi rilis Amazon EMR 5.0.0 dan yang lebih baru. Versi rilis sebelumnya termasuk Zeppelin sebagai suatu aplikasi sandbox. Untuk informasi selengkapnya, lihat Versi Amazon EMR rilis 4.x di Panduan Amazon EMR Rilis.

Secara default, Zeppelin dikonfigurasi dengan login dan kata sandi default yang tidak aman di lingkungan multi-penyewa.

Untuk mengkonfigurasi Zeppelin, selesaikan langkah-langkah berikut.

1. Memodifikasi mekanisme otentikasi.

   Ubah `shiro.ini` file untuk mengimplementasikan mekanisme otentikasi pilihan Anda. Zeppelin mendukung Direktori Aktif, LDAP, PAM, dan Knox SSO. Lihat autentikasi Apache Shiro untuk Apache Zeppelin untuk informasi selengkapnya.
2. Konfigurasikan Zeppelin untuk meniru pengguna akhir

   Ketika Anda mengizinkan Zeppelin untuk meniru pengguna akhir, pekerjaan yang dikirimkan oleh Zeppelin dapat dijalankan sebagai pengguna akhir tersebut. Tambahkan konfigurasi berikut `kecore-site.xml`:

```
[
  {
    "Classification": "core-site",
    "Properties": {
      "hadoop.proxyuser.zeppelin.hosts": "*",
      "hadoop.proxyuser.zeppelin.groups": "*"
    },
    "Configurations": [
    ]
  }
]
```

Selanjutnya, tambahkan konfigurasi berikut ke `hadoop-kms-site.xml` lokasi di`/etc/hadoop/conf`:

```
[
  {
    "Classification": "hadoop-kms-site",
    "Properties": {
      "hadoop.kms.proxyuser.zeppelin.hosts": "*",
      "hadoop.kms.proxyuser.zeppelin.groups": "*"
    },
    "Configurations": [
    ]
  }
]
```

Anda juga dapat menambahkan konfigurasi ini ke klaster EMR Amazon menggunakan konsol dengan mengikuti langkah-langkah dalam Mengkonfigurasi ulang grup instans di konsol.

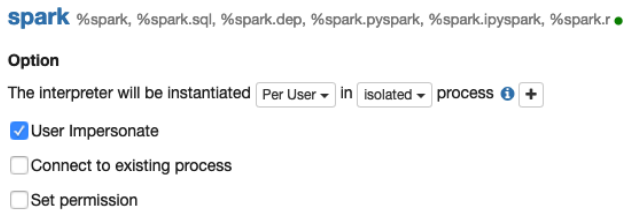
3. Izinkan Zeppelin ke sudo sebagai pengguna akhir

   Buat file `/etc/sudoers.d/90-zeppelin-user` yang berisi berikut ini:

   ```
   zeppelin ALL=(ALL) NOPASSWD:ALL
   ```

4. Ubah pengaturan interpreter untuk menjalankan tugas pengguna di proses mereka sendiri.

   Untuk semua penerjemah, konfigurasikan mereka untuk membuat instance penerjemah “Per Pengguna” dalam proses “terisolasi”.

5. Memodifikasi **`zeppelin-env.sh`**

   Tambahkan yang berikut ini `zeppelin-env.sh` agar Zeppelin mulai meluncurkan interpreter sebagai pengguna akhir:

   ```
   ZEPPELIN_IMPERSONATE_USER=`echo ${ZEPPELIN_IMPERSONATE_USER} | cut -d @ -f1`
   ```

```
export ZEPPELIN_IMPERSONATE_CMD='sudo -H -u ${ZEPPELIN_IMPERSONATE_USER} bash -c'
```

Tambahkan yang berikut ini `zeppelin-env.sh` untuk mengubah izin buku catatan default menjadi hanya-baca ke pembuat saja:

```
export ZEPPELIN_NOTEBOOK_PUBLIC="false"
```

Akhirnya, tambahkan berikut ini `zeppelin-env.sh` untuk menyertakan jalur RecordServer kelas EMR setelah pernyataan pertama`CLASSPATH`:

```
export CLASSPATH="$CLASSPATH:/usr/share/aws/emr/record-server/lib/aws-emr-record-
server-connector-common.jar:/usr/share/aws/emr/record-server/lib/aws-emr-record-
server-spark-connector.jar:/usr/share/aws/emr/record-server/lib/aws-emr-record-
server-client.jar:/usr/share/aws/emr/record-server/lib/aws-emr-record-server-
common.jar:/usr/share/aws/emr/record-server/lib/jars/secret-agent-interface.jar"
```

6. Mulai Ulang Zeppelin.

   Jalankan perintah berikut untuk me-restart Zeppelin:

   ```
   sudo systemctl restart zeppelin
   ```

Masalah yang diketahui

Masalah yang Diketahui

Ada masalah yang diketahui dalam Amazon EMR rilis 5.32 di mana izin untuk `hive-site.xml` diubah sehingga hanya pengguna istimewa yang dapat membacanya karena mungkin ada kredensyal yang disimpan di dalamnya. Ini dapat mencegah Hue membaca `hive-site.xml` dan menyebabkan halaman web terus dimuat ulang. Jika Anda mengalami masalah ini, Anda harus menambahkan konfigurasi berikut untuk memperbaiki masalah:

```
[
  {
    "Classification": "hue-ini",
    "Properties": {},
    "Configurations": [
      {
        "Classification": "desktop",
```

```
        "Properties": {
          "server_group":"hive_site_reader"
         },
        "Configurations":[
        ]
      }
    ]
  }
]
```

Ada masalah yang diketahui bahwa plugin EMRFS S3 untuk Apache Ranger saat ini tidak mendukung fitur Zona Keamanan Apache Ranger. Pembatasan kontrol akses yang ditentukan menggunakan fitur Zona Keamanan tidak diterapkan pada kluster EMR Amazon Anda.

UI Aplikasi

Secara default, Aplikasi UI tidak melakukan autentikasi. Ini termasuk ResourceManager UI, NodeManager UI, Livy UI, dan lainnya. Selain itu, setiap pengguna yang memiliki kemampuan untuk mengakses UI dapat melihat informasi tentang semua tugas pengguna lain.

Jika perilaku ini tidak diinginkan, Anda harus memastikan bahwa grup keamanan digunakan untuk membatasi akses ke aplikasi UI oleh pengguna.

Izin Default HDFS

Secara default, objek yang dibuat pengguna di HDFS diberikan izin baca dunia. Hal ini dapat berpotensi menyebabkan data yang dapat dibaca oleh pengguna yang seharusnya tidak memiliki akses ke sana. Untuk mengubah perilaku ini sehingga izin file default ditetapkan untuk baca dan tulis hanya oleh pembuat tugas, lakukan langkah-langkah berikut.

Saat membuat klaster EMR Anda, sediakan konfigurasi berikut:

```
[
  {
    "Classification": "hdfs-site",
    "Properties": {
      "dfs.namenode.acls.enabled": "true",
      "fs.permissions.umask-mode": "077",
      "dfs.permissions.superusergroup": "hdfsadmingroup"
    }
  }
```

```
]
```

Di samping itu, jalankan tindakan bootstrap berikut:

```
--bootstrap-actions Name='HDFS UMask Setup',Path=s3://elasticmapreduce/hdfs/umask/
umask-main.sh
```

Plugin Apache Ranger

Plugin Apache Ranger memvalidasi akses pengguna terhadap kebijakan otorisasi yang ditentukan dalam server admin kebijakan Apache Ranger.

Topik

- Plugin Apache Hive
- Plugin Apache Spark
- Plugin S3 EMRFS
- Plugin Trino

Plugin Apache Hive

Apache Hive adalah mesin eksekusi populer di ekosistem Hadoop. Amazon EMR menyediakan plugin Apache Ranger untuk dapat memberikan kendali akses lancar untuk Hive. Plugin ini kompatibel dengan server Admin Apache Ranger versi 2.0 dan versi terbaru.

Topik

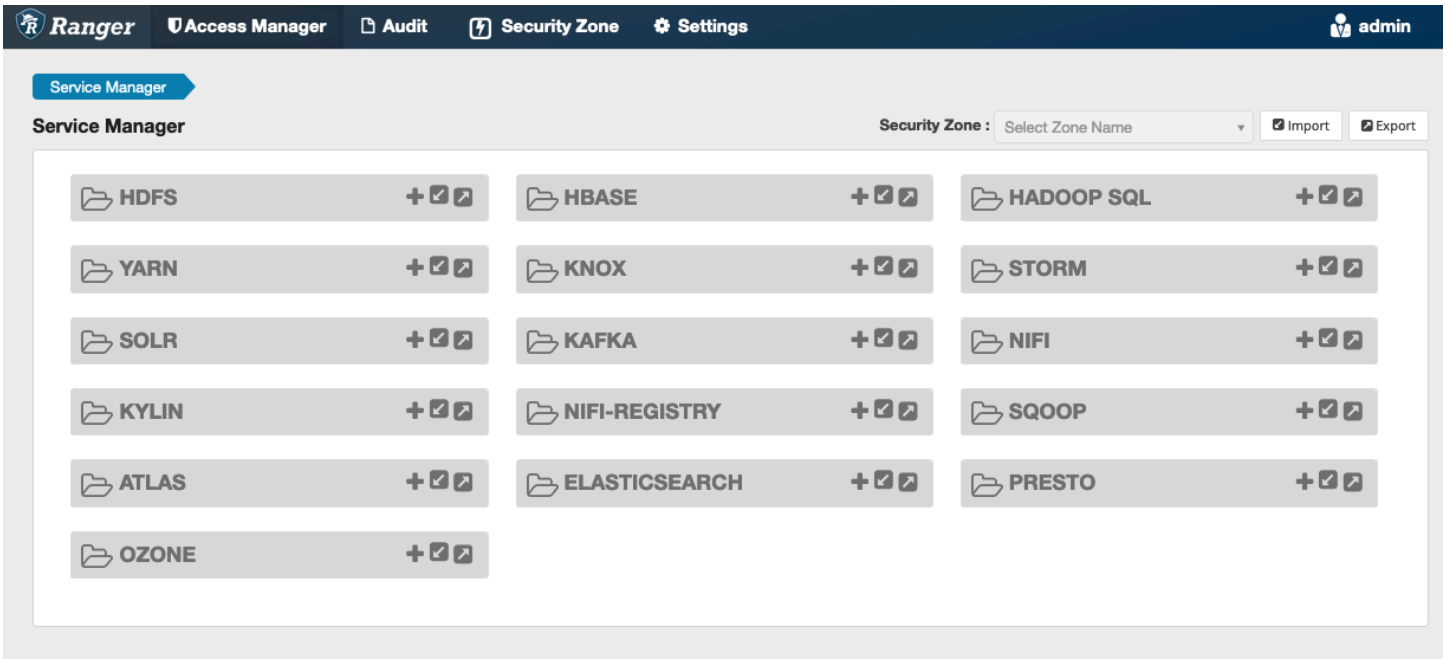
- Fitur yang didukung
- Instalasi konfigurasi layanan
- Pertimbangan-pertimbangan
- Batasan

Fitur yang didukung

Plugin Apache Ranger untuk Hive di EMR mendukung semua fungsi dari plugin sumber terbuka, yang meliputi basis data, tabel, kendali akses tingkat kolom dan penyaringan baris dan masking data. Untuk tabel perintah Hive dan terkait izin Ranger, lihat perintah Hive untuk pemetaan izin Ranger.

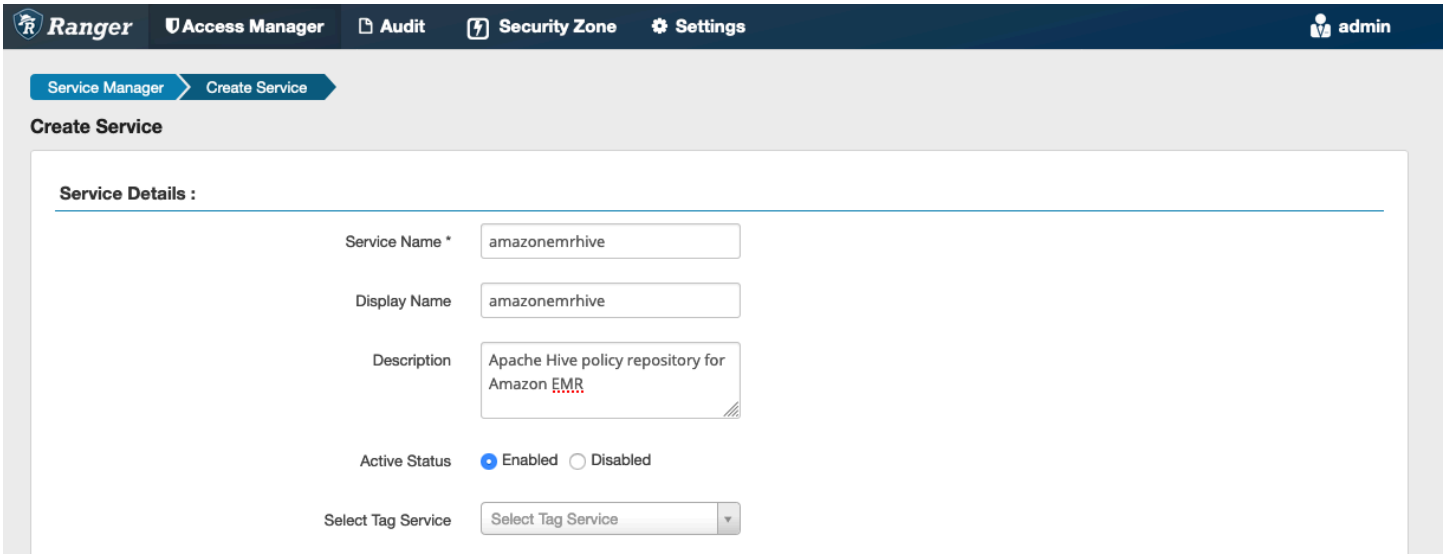
Instalasi konfigurasi layanan

Plugin Apache Hive kompatibel dengan definisi layanan Hive yang ada dalam Apache Hive Hadoop SQL.



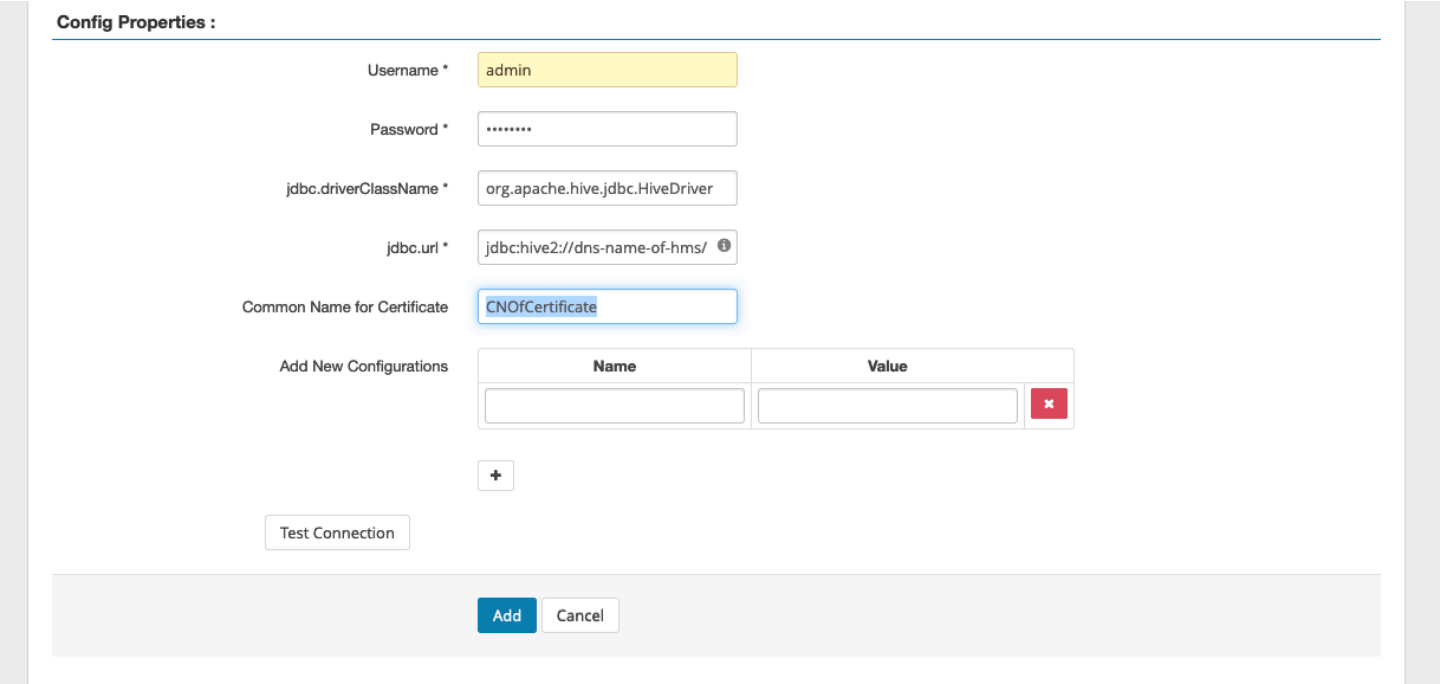
Jika Anda tidak memiliki sebuah instans dari layanan di bawah Hadoop SQL, seperti yang ditunjukkan di atas, Anda dapat membuat satu. Klik pada + di sebelah Hadoop SQL.

1. Nama Layanan (Jika ditampilkan): Anda perlu memasukkan nama layanan. Nilai yang direkomendasikan adalah **`amazonemrhive`**. Buat catatan nama layanan ini -- itu akan dibutuhkan saat membuat konfigurasi keamanan EMR.
2. Nama Tampilan: Memasukkan nama yang akan ditampilkan untuk layanan. Nilai yang direkomendasikan adalah **`amazonemrhive`**.

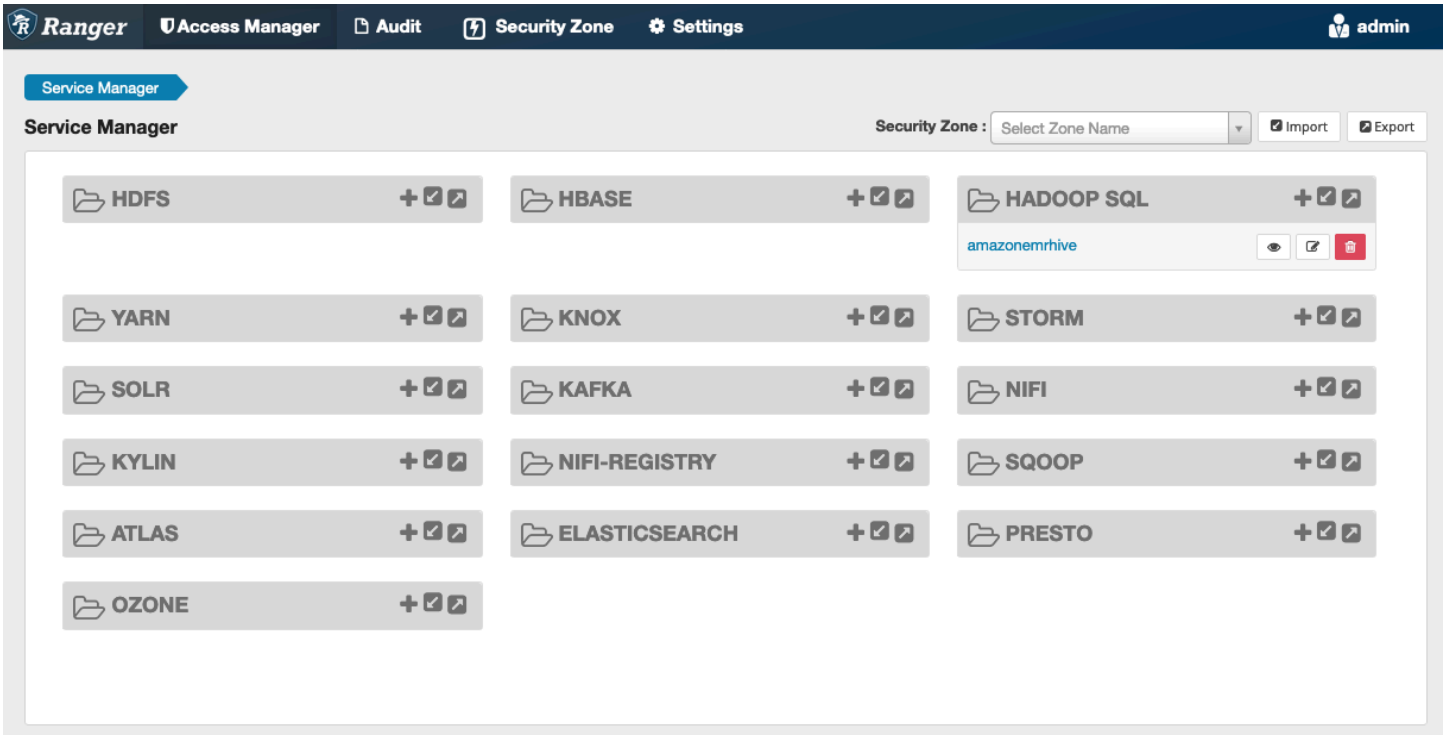


Properti Apache Hive Config digunakan untuk membuat koneksi ke server Admin Apache Ranger Anda dengan 2 untuk mengimplementasikan auto complete saat HiveServer membuat kebijakan. Properti di bawah ini tidak diharuskan akurat jika Anda tidak memiliki proses HiveServer 2 yang persisten dan dapat diisi dengan informasi apa pun.

- Username: Masukkan nama pengguna untuk koneksi JDBC ke instance instance HiveServer 2.
- Kata Sandi: Anda perlu memasukkan kata sandi untuk nama pengguna di atas.
- jdbc.driver. ClassName: Masukkan nama kelas kelas JDBC untuk konektivitas Apache Hive. Nilai default dapat digunakan.
- jdbc.url: Masukkan string koneksi JDBC yang akan digunakan saat menghubungkan ke 2. HiveServer
- Nama Umum untuk Sertifikat: Bidang CN di sertifikat yang digunakan untuk connect ke server admin dari plugin klien. Nilai ini harus cocok dengan bidang CN di sertifikat TLS Anda yang dibuat untuk plugin.



Tombol Test Connection menguji apakah nilai-nilai di atas dapat digunakan untuk berhasil terhubung ke instance HiveServer 2. Setelah layanan berhasil dibuat, Manajer Layanan akan terlihat seperti di bawah ini:



Pertimbangan-pertimbangan

Server metadata sarang

Server metadata Hive hanya dapat diakses oleh mesin tepercaya, khususnya Hive dan`emr_record_server`, untuk melindungi dari akses yang tidak sah. Server metadata Hive juga diakses oleh semua node di cluster. Port 9083 yang diperlukan menyediakan semua node akses ke node utama.

Otentikasi

Secara default, Apache Hive dikonfigurasi untuk mengautentikasi menggunakan Kerberos seperti yang dikonfigurasi dalam konfigurasi EMR Security. HiveServer2 dapat dikonfigurasi untuk mengautentikasi pengguna menggunakan LDAP juga. Lihat Menerapkan autentikasi LDAP untuk Hive pada multi-penyewa klaster Amazon EMR untuk informasi.

Batasan

Berikut ini adalah batasan saat ini untuk plugin Apache Hive di Amazon EMR 5.x:

- Peran Hive saat ini tidak didukung. Pernyataan Berikan, Batalkan tidak didukung.
- Hive CLI tidak didukung. JDBC/Beeline adalah satu-satunya cara yang diotorisasi untuk connect ke Hive.
- Konfigurasi `hive.server2.builtin.udf.blacklist` harus diisi dengan UDFS yang Anda anggap tidak aman.

Plugin Apache Spark

Amazon EMR telah mengintegrasikan EMR RecordServer untuk menyediakan kontrol akses berbutir halus untuk SparkSQL. EMR RecordServer adalah proses istimewa yang berjalan di semua node pada cluster yang mendukung Apache Ranger. Ketika driver atau eksekutor Spark menjalankan pernyataan SparkSQL, semua metadata dan permintaan data akan melalui. RecordServer Untuk mempelajari lebih lanjut tentang EMR RecordServer, lihat halaman. Komponen Amazon EMR

Topik

- Fitur yang didukung
- Menerapkan kembali definisi layanan untuk menggunakan pernyataan INSERT, ALTER, atau DDL
- Instalasi definisi layanan
- Membuat kebijakan SparkSQL

- Pertimbangan-pertimbangan
- Batasan

Fitur yang didukung

Pernyataan SQL/Tindakan Ranger	STATUS	Rilis EMR yang didukung
PILIH	Didukung	Pada 5.32
TAMPILKAN BASIS DATA	Didukung	Pada 5.32
TAMPILKAN KOLOM	Didukung	Pada 5.32
TAMPILKAN TABEL	Didukung	Pada 5.32
TAMPILKAN PROPERTI TABEL	Didukung	Pada 5.32
GAMBAR TABEL	Didukung	Pada 5.32
SISIPKAN TIMPA	Didukung	Pada 5,34 dan 6,4
MASUKKAN KE	Didukung	Pada 5,34 dan 6,4
ALTER TABLE	Didukung	Pada 6.4
CREATE TABLE	Didukung	Pada 5,35 dan 6,7
BUAT BASIS DATA	Didukung	Pada 5,35 dan 6,7
MEJA DROP	Didukung	Pada 5,35 dan 6,7

Pernyataan SQL/Tindakan Ranger	STATUS	Rilis EMR yang didukung
DROP DATABASE	Didukung	Pada 5,35 dan 6,7
TAMPILAN DROP	Didukung	Pada 5,35 dan 6,7
BUAT TAMPILAN	Tidak Didukung	

Fitur berikut didukung saat menggunakan SparkSQL:

- Kendali akses lancar pada tabel di Metastore Hive, dan kebijakan dapat dibuat pada tingkat basis data, tabel, dan kolom.
- Kebijakan Apache Ranger dapat mencakup kebijakan hibah dan tolak kebijakan untuk pengguna dan grup.
- Acara audit diserahkan ke CloudWatch Log.

Menerapkan kembali definisi layanan untuk menggunakan pernyataan INSERT, ALTER, atau DDL

> **Note**
>
> Dimulai dengan Amazon EMR 6.4, Anda dapat menggunakan Spark SQL dengan pernyataan: INSERT INTO, INSERT OVERWRITE, atau ALTER TABLE. Dimulai dengan Amazon EMR 6.7, Anda dapat menggunakan Spark SQL untuk membuat atau menjatuhkan database dan tabel. Jika Anda memiliki instalasi yang ada di server Apache Ranger dengan definisi layanan Apache Spark digunakan, gunakan kode berikut untuk menerapkan definisi layanan.
>
> ```
> # Get existing Spark service definition id calling Ranger REST API and JSON
>  processor
> curl --silent -f -u <admin_user_login>:<password_for_ranger_admin_user> \
> -H "Accept: application/json" \
> -H "Content-Type: application/json" \
> -k 'https://*<RANGER SERVER ADDRESS>*:6182/service/public/v2/api/servicedef/
> name/amazon-emr-spark' | jq .id
> ```

```
# Download the latest Service definition
wget https://s3.amazonaws.com/elasticmapreduce/ranger/service-definitions/
version-2.0/ranger-servicedef-amazon-emr-spark.json

# Update the service definition using the Ranger REST API
curl -u <admin_user_login>:<password_for_ranger_admin_user> -X PUT -d @ranger-
servicedef-amazon-emr-spark.json \
-H "Accept: application/json" \
-H "Content-Type: application/json" \
-k 'https://*<RANGER SERVER ADDRESS>*:6182/service/public/v2/api/
servicedef/<Spark service definition id from step 1>'
```

Instalasi definisi layanan

Instalasi defenisi layanan EMR Apache Spark memerlukan server Admin Ranger untuk setup. Lihat Atur server Admin Ranger.

Ikuti langkah-langkah untuk menginstal definisi layanan Apache Spark:

Langkah 1: SSH ke server Admin Apache Ranger

Misalnya:

```
ssh ec2-user@ip-xxx-xxx-xxx-xxx.ec2.internal
```

Langkah 2: Unduh definisi layanan dan plugin server Admin Apache Ranger

Di direktori sementara, unduh definisi layanan. Definisi layanan ini didukung oleh versi Ranger 2.x.

```
mkdir /tmp/emr-spark-plugin/
cd /tmp/emr-spark-plugin/

wget https://s3.amazonaws.com/elasticmapreduce/ranger/service-definitions/version-2.0/
ranger-spark-plugin-2.x.jar
wget https://s3.amazonaws.com/elasticmapreduce/ranger/service-definitions/version-2.0/
ranger-servicedef-amazon-emr-spark.json
```

Langkah 3: Instal plugin Apache Spark untuk Amazon EMR

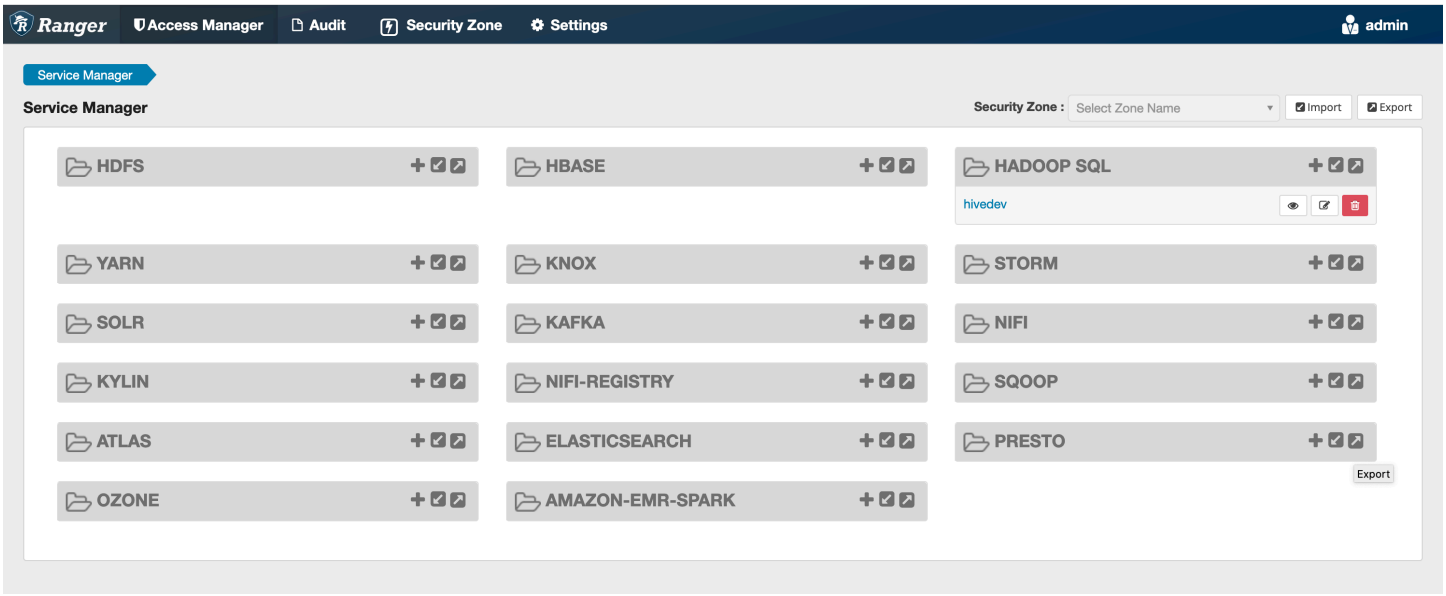
```
export RANGER_HOME=.. # Replace this Ranger Admin's home directory eg /usr/lib/ranger/
ranger-2.0.0-admin
```

```
mkdir $RANGER_HOME/ews/webapp/WEB-INF/classes/ranger-plugins/amazon-emr-spark
mv ranger-spark-plugin-2.x.jar $RANGER_HOME/ews/webapp/WEB-INF/classes/ranger-plugins/
amazon-emr-spark
```

Langkah 4: Daftarkan definisi layanan Apache Spark untuk Amazon EMR

```
curl -u *<admin users login>*:*_<_**_password_ **_for_** _ranger admin user_**_>_* -X
 POST -d @ranger-servicedef-amazon-emr-spark.json \
-H "Accept: application/json" \
-H "Content-Type: application/json" \
-k 'https://*<RANGER SERVER ADDRESS>*:6182/service/public/v2/api/servicedef'
```

Jika perintah ini berhasil, Anda akan melihat layanan baru di Anda UI Admin Ranger yang disebut "AMAZON-EMR-SPARK", seperti yang ditunjukkan pada citra berikut (Ranger versi 2.0 ditampilkan).

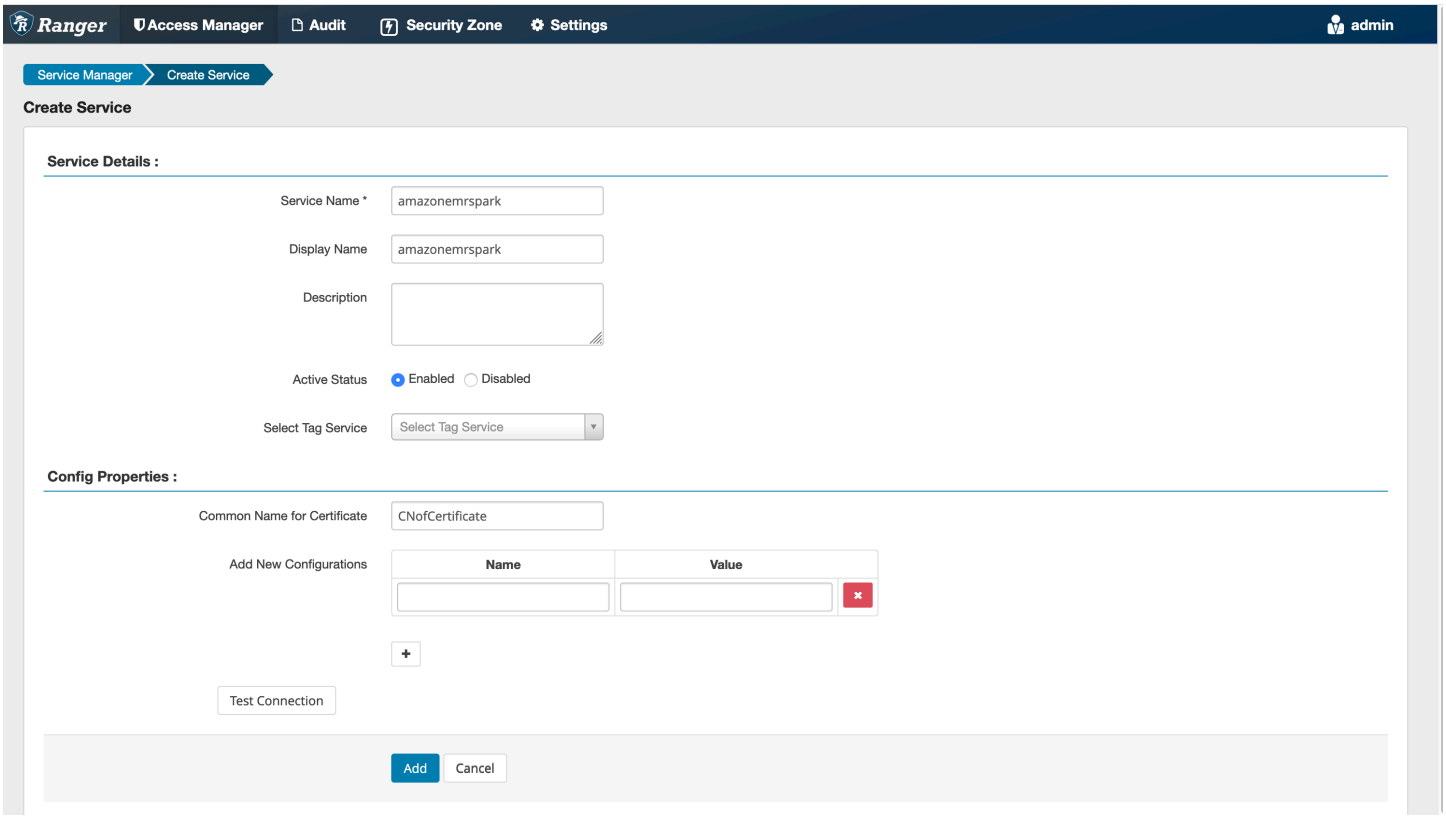


Langkah 5: Buat instance aplikasi AMAZON-EMR-SPARK

Nama Layanan (Jika ditampilkan): Nama layanan yang akan digunakan. Nilai yang direkomendasikan adalah **`amazonemrspark`**. Catat nama layanan ini saat membuat konfigurasi keamanan EMR.

Nama tampilan: Nama yang akan ditampilkan untuk instans ini. Nilai yang direkomendasikan adalah **`amazonemrspark`**.

Nama Umum Untuk Sertifikat: Bidang CN di sertifikat yang digunakan untuk connect ke server admin dari plugin klien. Nilai ini harus cocok dengan bidang CN di sertifikat TLS Anda yang dibuat untuk plugin.



> **Note**
>
> Sertifikat TLS untuk plugin ini harus telah terdaftar di penyimpanan kepercayaan pada server Admin Ranger. Lihat Sertifikat TLS untuk detail selengkapnya.

Membuat kebijakan SparkSQL

Saat membuat kebijakan baru, bidang yang harus diisi adalah:

Nama Kebijakan: Nama kebijakan ini.

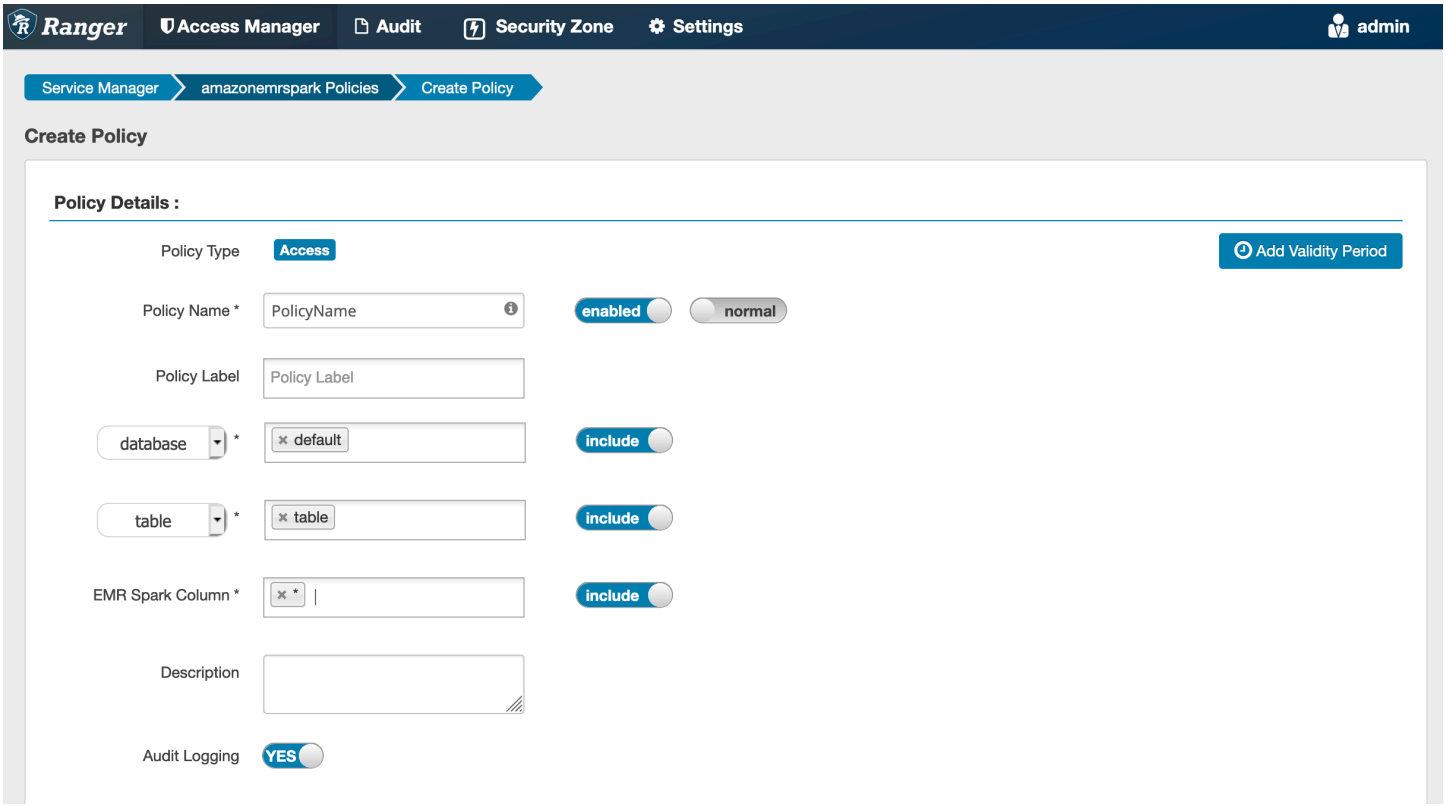
Label Kebijakan: Label yang dapat Anda tempatkan di kebijakan ini.

Basis data: Basis data yang berlaku untuk kebijakan ini. Wildcard “*” mewakili semua database.

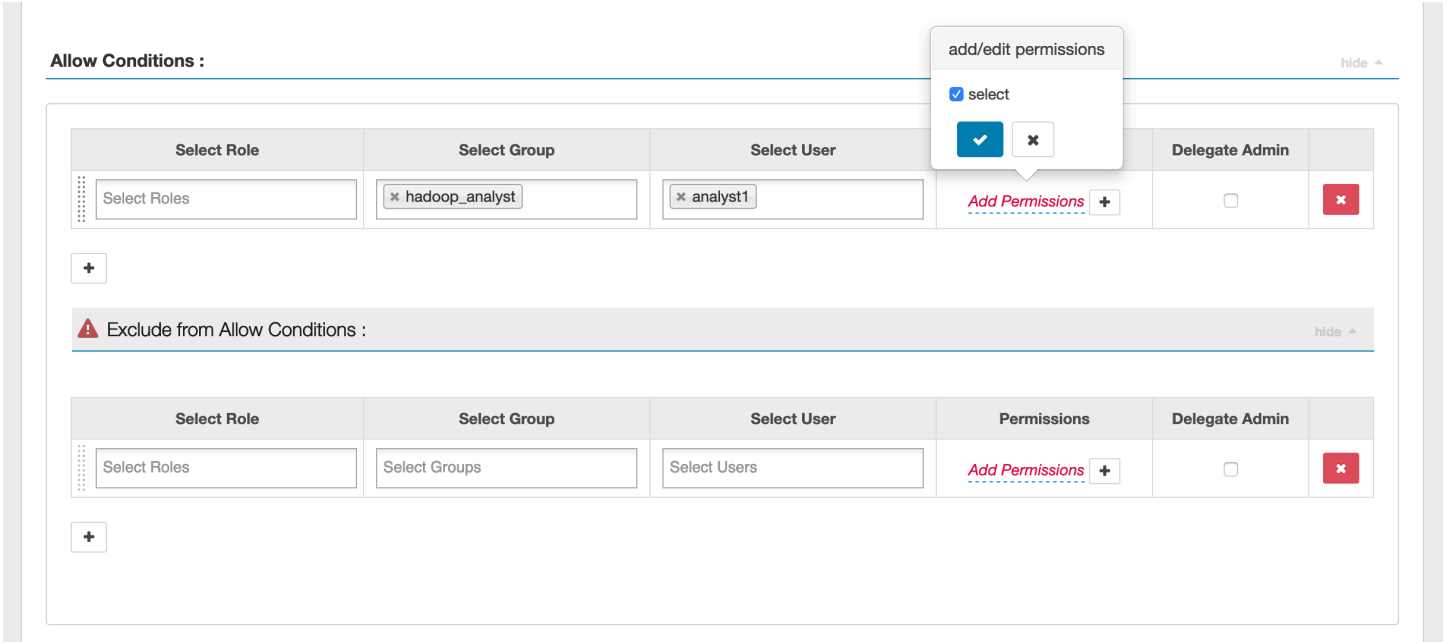
Tabel: Tabel yang berlaku untuk kebijakan ini. Wildcard “*” mewakili semua tabel.

Kolom EMR Spark: Kolom yang berlaku untuk kebijakan ini. Wildcard "*" mewakili semua kolom.

Deskripsi: Deskripsi dari kebijakan ini.



Untuk menentukan pengguna dan grup, Anda perlu memasukkan pengguna dan grup di bawah ini untuk memberikan izin. Anda juga dapat menentukan pengecualian untuk kondisi izinkan dan kondisi penolakan.



Setelah menentukan mengizinkan dan tolak syarat, klik Simpan.

Pertimbangan-pertimbangan

Setiap node dalam cluster EMR harus dapat terhubung ke node utama pada port 9083.

Batasan

Berikut ini adalah batasan saat ini untuk plugin Apache Spark:

- Server Catatan akan selalu terhubung ke HMS yang berjalan pada klaster Amazon EMR. Konfigurasikan HMS untuk connect ke Mode Jarak Jauh, jika diperlukan. Anda tidak harus config nilai-nilai di file konfigurasi Apache Spark Hive-site.xml.
- Tabel yang dibuat menggunakan sumber data Spark pada CSV atau Avro tidak dapat dibaca menggunakan EMR. RecordServer Gunakan Hive untuk membuat dan tulis data, dan membaca menggunakan Catatan.
- Tabel Delta Lake dan Hudi tidak didukung.
- Pengguna harus memiliki akses ke basis data default. Ini adalah persyaratan untuk Apache Spark.
- Server Admin Ranger tidak support selesai otomatis.
- Plugin SparkSQL untuk Amazon EMR tidak support filter baris atau masking data.
- Saat menggunakan ALTER TABLE dengan Spark SQL, lokasi partisi harus menjadi direktori anak dari lokasi tabel. Memasukkan data ke dalam partisi di mana lokasi partisi berbeda dari lokasi tabel tidak didukung.

Plugin S3 EMRFS

Untuk membuatnya lebih mudah untuk menyediakan kontrol akses terhadap objek di S3 pada cluster multi-tenant, plugin EMRFS S3 menyediakan kontrol akses ke data dalam S3 saat mengaksesnya melalui EMRFS. Anda dapat mengizinkan akses ke sumber daya S3 pada tingkat pengguna dan grup.

Untuk mencapai hal ini, ketika aplikasi Anda mencoba untuk mengakses data di S3, EMRFS mengirimkan permintaan untuk kredensial dalam proses Agen Rahasia, di mana permintaan diautentikasi dan diotorisasi terhadap plugin Apache Ranger. Jika permintaan diotorisasi, maka agen rahasia mengambil IAM role untuk Mesin Apache Ranger dengan kebijakan terbatas untuk menghasilkan kredensial yang hanya memiliki akses ke kebijakan Ranger yang mengizinkan akses. Kredensialnya kemudian diteruskan kembali ke EMRFS untuk mengakses S3.

Topik

- Fitur yang didukung

- Instalasi konfigurasi layanan
- Membuat kebijakan S3 EMRFS
- Catatan penggunaan kebijakan S3 EMRFS
- Batasan

Fitur yang didukung

Plugin EMRFS S3 menyediakan otorisasi tingkat penyimpanan. Kebijakan dapat dibuat untuk menyediakan akses ke pengguna dan grup ke bucket S3 dan prefiks. Otorisasi dilakukan hanya terhadap EMRFS.

Instalasi konfigurasi layanan

Instalasi definisi layanan Trino mengharuskan server Admin Ranger diatur. Untuk mengatur pemisah Admin Ranger, lihat. Atur server Admin Ranger

Ikuti langkah berikut untuk menginstal definisi layanan EMRFS.

Langkah 1: SSH ke server Admin Apache Ranger.

Misalnya:

```
ssh ec2-user@ip-xxx-xxx-xxx-xxx.ec2.internal
```

Langkah 2: Unduh definisi layanan Amazon EMR dan plugin server Admin Apache Ranger.

Di direktori sementara, unduh definisi layanan Amazon EMR. Definisi layanan ini didukung oleh versi Ranger 2.x.

```
mkdir /tmp/emr-emrfs-plugin/
cd /tmp/emr-emrfs-plugin/

wget https://s3.amazonaws.com/elasticmapreduce/ranger/service-definitions/version-2.0/
ranger-servicedef-amazon-emr-emrfs.json
wget https://s3.amazonaws.com/elasticmapreduce/ranger/service-definitions/version-2.0/
ranger-emr-emrfs-plugin-2.x.jar
```

Langkah 3: Instal plugin S3 EMRFS Apache untuk Amazon EMR.

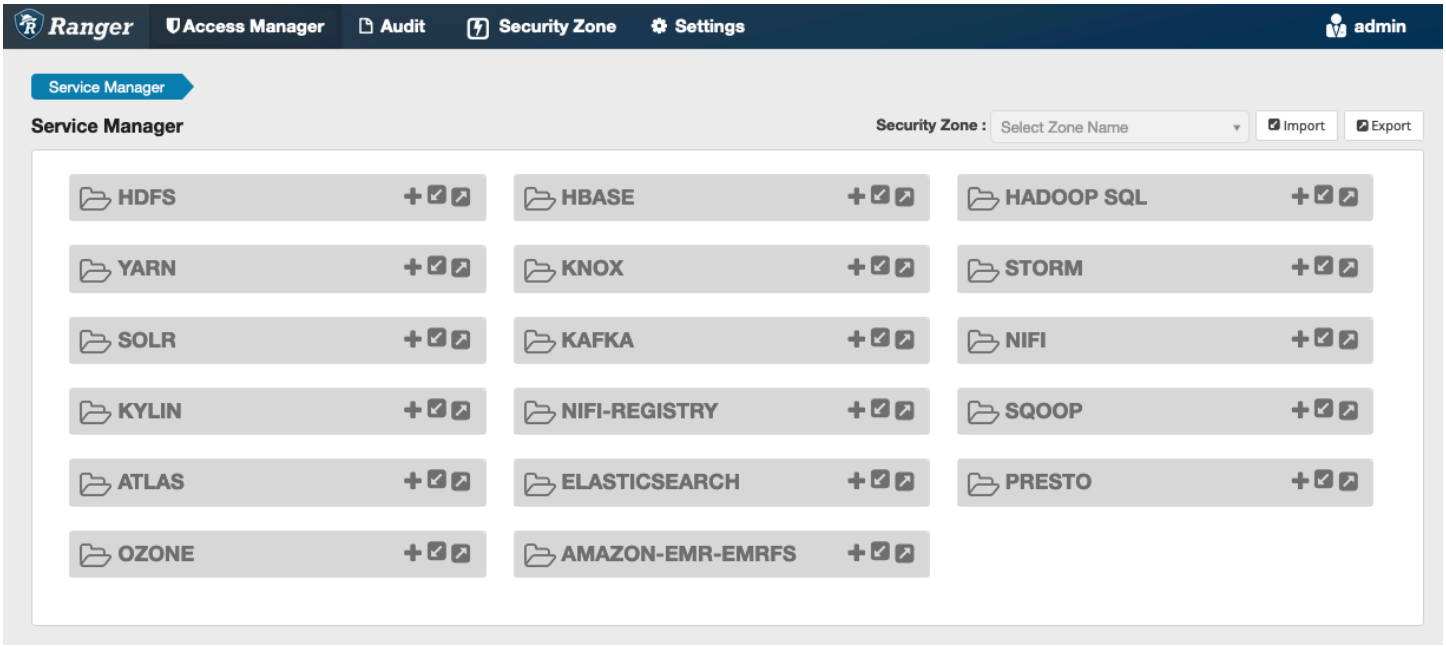
```
export RANGER_HOME=.. # Replace this Ranger Admin's home directory eg /usr/lib/ranger/
ranger-2.0.0-admin
```

```
mkdir $RANGER_HOME/ews/webapp/WEB-INF/classes/ranger-plugins/amazon-emr-emrfs
mv ranger-emr-emrfs-plugin-2.x.jar $RANGER_HOME/ews/webapp/WEB-INF/classes/ranger-
plugins/amazon-emr-emrfs
```

Langkah 4: Daftarkan definisi layanan S3 EMRFS.

```
curl -u *<admin users login>*:*_<_**_password_ **_for_** _ranger admin user_**_>_* -X
 POST -d @ranger-servicedef-amazon-emr-emrfs.json \
-H "Accept: application/json" \
-H "Content-Type: application/json" \
-k 'https://*<RANGER SERVER ADDRESS>*:6182/service/public/v2/api/servicedef'
```

Jika perintah ini berhasil, Anda melihat layanan baru di UI Admin Ranger yang disebut "AMAZON-EMR-S3", seperti yang ditunjukkan pada citra berikut (Ranger versi 2.0 ditampilkan).



Langkah 5: Buat sebuah instans dari aplikasi AMAZON-EMR-EMRFS.

Buat sebuah instans dari definisi layanan.

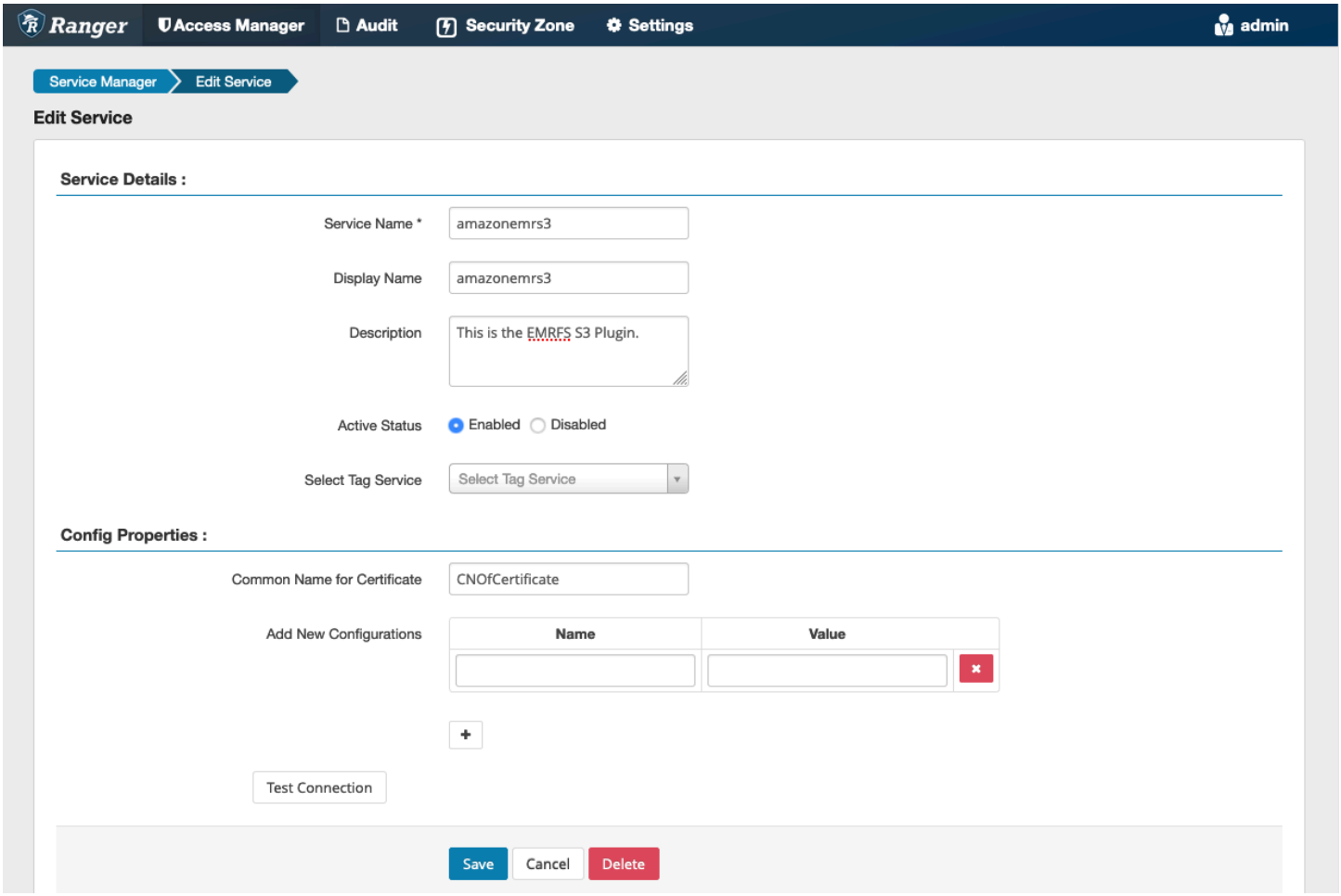
- Klik pada + di sebelah AMAZON-EMR-EMRFS.

Isi kolom berikut:

Nama Layanan (Jika ditampilkan): Nilai yang direkomendasikan adalah **`amazonemrspark`**. Catat nama layanan ini saat membuat konfigurasi keamanan EMR.

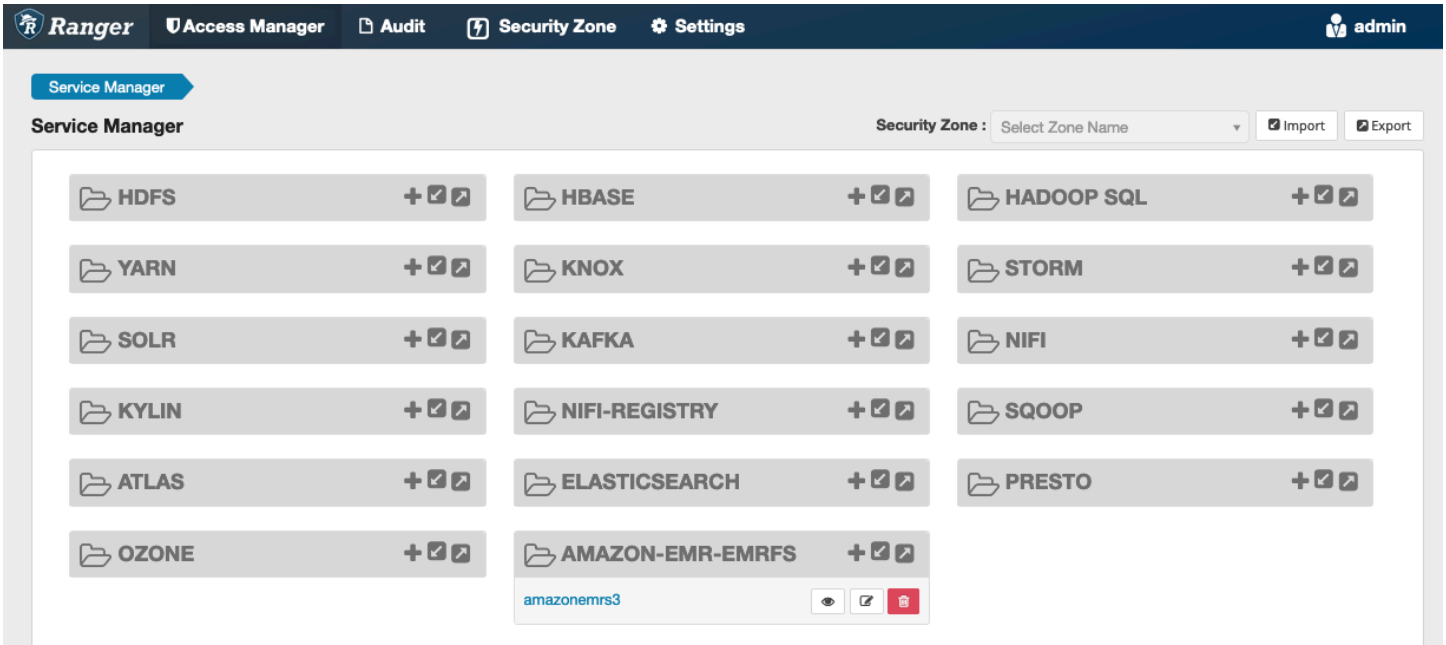
Nama Tampilan: Nama yang akan ditampilkan untuk layanan. Nilai yang direkomendasikan adalah **`amazonemrspark`**.

Nama Umum Untuk Sertifikat: Bidang CN di sertifikat yang digunakan untuk connect ke server admin dari plugin klien. Nilai ini harus cocok dengan bidang CN dalam sertifikat TLS yang dibuat untuk plugin.



> **Note**
>
> Sertifikat TLS untuk plugin ini harus telah terdaftar di penyimpanan kepercayaan pada server Admin Ranger. Lihat Sertifikat TLS untuk detail selengkapnya.

Ketika layanan dibuat, Manajer Service termasuk "AMAZON-EMR-EMRFS", seperti yang ditunjukkan pada citra berikut.



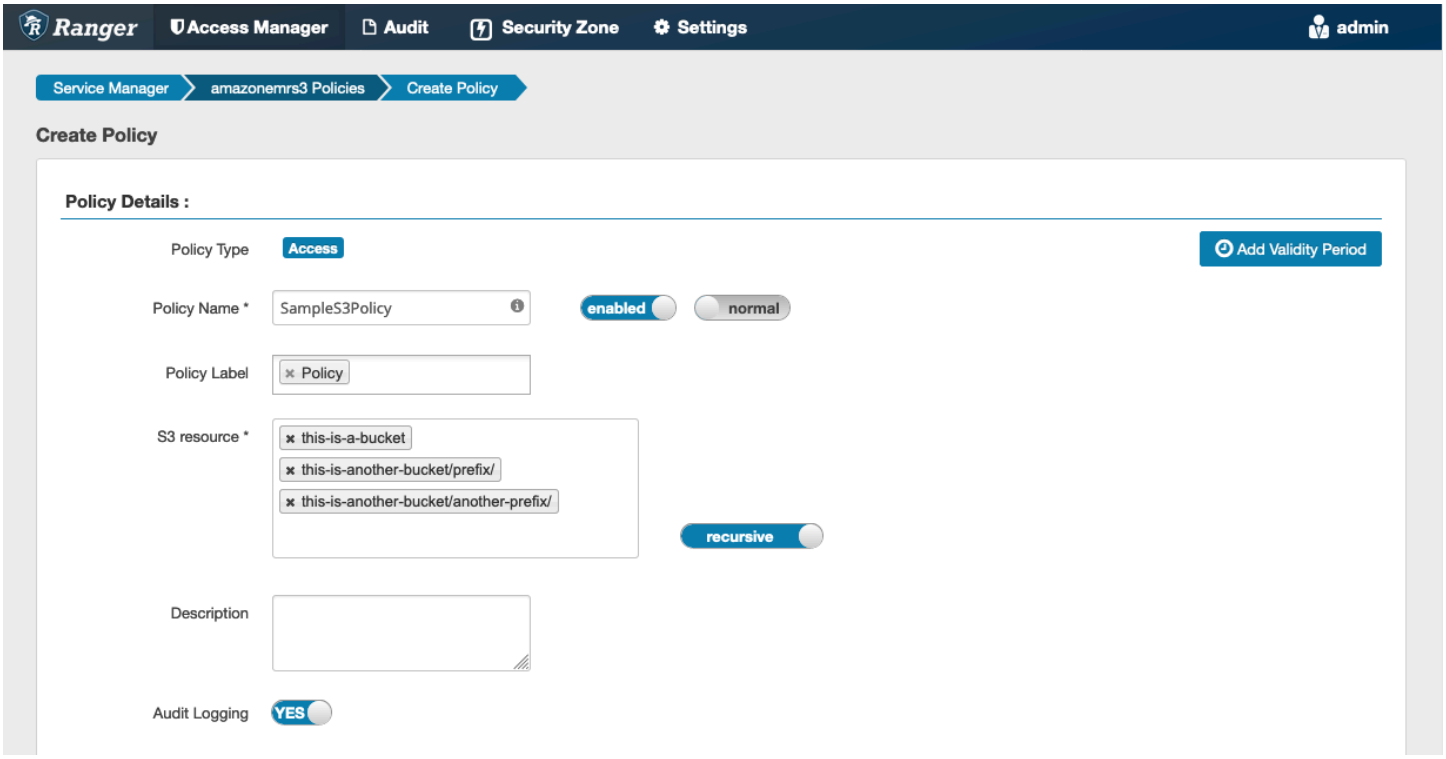
Membuat kebijakan S3 EMRFS

Untuk membuat kebijakan baru di halaman Buat kebijakan Manajer Layanan, isi kolom berikut.

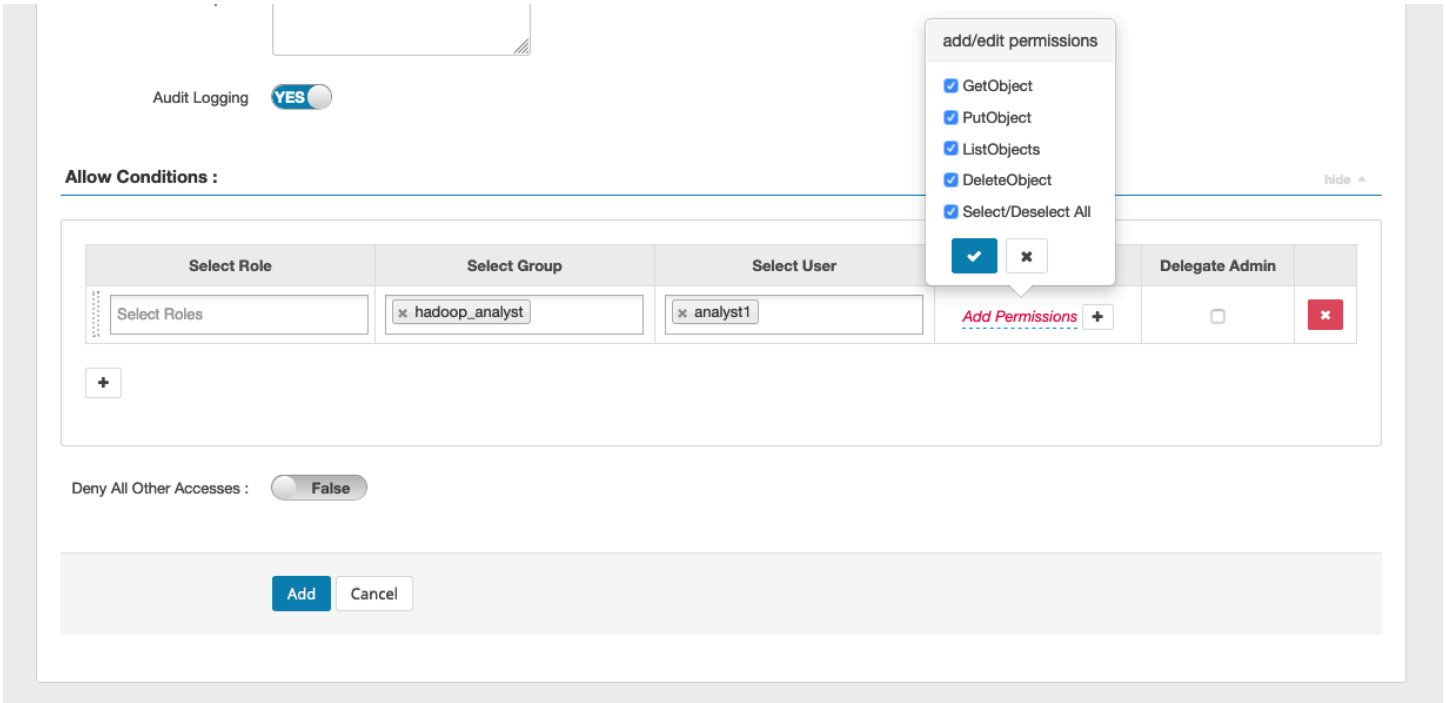
Nama Kebijakan: Nama kebijakan ini.

Label Kebijakan: Label yang dapat Anda tempatkan di kebijakan ini.

Sumber Daya S3: Sumber daya yang dimulai dengan bucket dan prefiks opsional. Lihat Catatan penggunaan kebijakan S3 EMRFS untuk informasi tentang praktik terbaik. Sumber daya di server Admin Ranger tidak boleh berisi **`s3://`**, **`s3a://`** atau **`s3n://`**.



Anda dapat menentukan pengguna dan grup untuk memberikan izin. Anda juga dapat menentukan pengecualian untuk kondisi izinkan dan kondisi penolakan.



> **Note**
>
> Maksimum tiga sumber daya diperbolehkan untuk setiap kebijakan. Menambahkan lebih dari tiga sumber daya dapat mengakibatkan kesalahan ketika kebijakan ini digunakan pada klaster EMR. Menambahkan lebih dari tiga kebijakan akan menampilkan pengingat tentang batas kebijakan.

Catatan penggunaan kebijakan S3 EMRFS

Saat membuat kebijakan S3 di Apache Ranger, ada beberapa pertimbangan penggunaan yang harus diperhatikan.

Izin untuk beberapa objek S3

Anda dapat menggunakan kebijakan rekursif dan ekspresi wildcard untuk memberikan izin untuk beberapa objek S3 dengan prefiks umum. Kebijakan rekursif memberikan izin untuk semua objek dengan prefiks umum. Ekspresi wildcard memilih beberapa prefiks. Bersama-sama, mereka memberikan izin ke semua objek dengan beberapa prefiks umum seperti yang ditunjukkan di contoh berikut.

Example Menggunakan kebijakan rekursif

Misalkan Anda ingin izin untuk daftar semua file parket di bucket S3 seperti yang diorganisir sebagai berikut.

```
s3://sales-reports/americas/
    +- year=2000
    |      +- data-q1.parquet
    |      +- data-q2.parquet
    +- year=2019
    |      +- data-q1.json
    |      +- data-q2.json
    |      +- data-q3.json
    |      +- data-q4.json
    |
    +- year=2020
    |      +- data-q1.parquet
    |      +- data-q2.parquet
    |      +- data-q3.parquet
    |      +- data-q4.parquet
```

```
|      +- annual-summary.parquet
+- year=2021
```

Pertama, pertimbangkan file parket dengan prefiks `s3://sales-reports/americas/year=2000`. Anda dapat memberikan GetObject izin untuk semuanya dengan dua cara:

Menggunakan kebijakan non-rekursif: salah satu pilihan adalah dengan menggunakan dua kebijakan non-rekursif terpisah, satu untuk direktori dan yang lainnya untuk file.

Kebijakan pertama memberikan izin ke prefiks `s3://sales-reports/americas/year=2020` (tidak ada penjejakan /).

```
- S3 resource = "sales-reports/americas/year=2000"
- permission = "GetObject"
- user = "analyst"
```

Kebijakan kedua menggunakan ekspresi wildcard untuk memberikan izin semua file dengan prefiks `sales-reports/americas/year=2020/`(perhatikan penjejakan /).

```
- S3 resource = "sales-reports/americas/year=2020/*"
- permission = "GetObject"
- user = "analyst"
```

Menggunakan kebijakan rekursif: Alternatif yang lebih nyaman adalah dengan menggunakan kebijakan rekursif tunggal dan memberikan izin rekursif untuk prefiks.

```
- S3 resource = "sales-reports/americas/year=2020"
- permission = "GetObject"
- user = "analyst"
- is recursive = "True"
```

Sejauh ini, hanya file parket dengan prefiks `s3://sales-reports/americas/year=2000` yang telah dimasukkan. Anda sekarang dapat juga menyertakan file parket dengan prefiks yang berbeda, `s3://sales-reports/americas/year=2020`, ke kebijakan rekursif yang sama dengan memperkenalkan ekspresi wildcard sebagai berikut.

```
- S3 resource = "sales-reports/americas/year=20?0"
- permission = "GetObject"
- user = "analyst"
- is recursive = "True"
```

Kebijakan untuk PutObject dan DeleteObject izin

Menulis kebijakan untuk `PutObject` dan `DeleteObject` izin ke file di EMRFS memerlukan perhatian khusus karena, tidak seperti GetObject izin, mereka memerlukan izin rekursif tambahan yang diberikan ke awalan.

Example Kebijakan untuk PutObject dan DeleteObject izin

Misalnya, menghapus file tidak hanya `annual-summary.parquet` memerlukan DeleteObject izin ke file yang sebenarnya.

```
- S3 resource = "sales-reports/americas/year=2020/annual-summary.parquet"
- permission = "DeleteObject"
- user = "analyst"
```

Hal ini juga membutuhkan kebijakan pemberian rekursif `GetObject` dan `PutObject` hak istimewa ke prefiksnya.

Demikian pula, memodifikasi file `annual-summary.parquet`, membutuhkan tidak hanya izin `PutObject` untuk file yang sebetulnya.

```
- S3 resource = "sales-reports/americas/year=2020/annual-summary.parquet"
- permission = "PutObject"
- user = "analyst"
```

Hal ini juga membutuhkan izin `GetObject` pemberian kebijakan untuk prefiksnya.

```
- S3 resource = "sales-reports/americas/year=2020"
- permission = "GetObject"
- user = "analyst"
- is recursive = "True"
```

Wildcard di kebijakan

Ada dua wilayah di mana wildcard dapat ditentukan. Saat menentukan sumber daya S3, "*" dan "?" dapat digunakan. "*" menyediakan pencocokan terhadap jalur S3 dan cocok dengan segala sesuatu setelah prefiks. Misalnya, lihat kebijakan berikut ini.

```
S3 resource = "sales-reports/americas/*"
```

Ini cocok dengan jalur S3 berikut.

```
sales-reports/americas/year=2020/
sales-reports/americas/year=2019/
sales-reports/americas/year=2019/month=12/day=1/afile.parquet
sales-reports/americas/year=2018/month=6/day=1/afile.parquet
sales-reports/americas/year=2017/afile.parquet
```

Wilcard "?" cocok hanya satu karakter. Misalnya, untuk kebijakan.

```
S3 resource = "sales-reports/americas/year=201?/"
```

Ini cocok dengan jalur S3 berikut.

```
sales-reports/americas/year=2019/
sales-reports/americas/year=2018/
sales-reports/americas/year=2017/
```

Wildcard di pengguna

Ada dua wildcard built-in saat menetapkan pengguna untuk menyediakan akses ke pengguna. Yang pertama adalah wildcard "{PENGGUNA}" yang menyediakan akses ke semua pengguna. Wildcard kedua adalah "{PEMILIK}", yang menyediakan akses kepada pemilik objek tertentu atau secara langsung. Namun, wildcard "{PENGGUNA}" saat ini tidak didukung.

Batasan

Berikut ini adalah batasan saat ini dari plugin EMRFS S3:

- Kebijakan Apache Ranger dapat memiliki maksimal tiga kebijakan.
- Akses ke S3 harus dilakukan melalui EMRFS dan dapat digunakan dengan aplikasi terkait Hadoop. Berikut ini tidak didukung:

  - Perpustakaan Boto3

  - AWS SDK dan AWK CLI

  - Penyambung sumber terbuka S3A
- Apache Ranger tolak kebijakan tidak didukung.

- Operasi pada S3 dengan kunci yang memiliki enkripsi CSE-KMS saat ini tidak didukung.
- Support lintas wilayah tidak didukung.
- Fitur Zona Keamanan Apache Ranger tidak didukung. Pembatasan kontrol akses yang ditentukan menggunakan fitur Zona Keamanan tidak diterapkan pada kluster EMR Amazon Anda.
- Pengguna Hadoop tidak menghasilkan peristiwa audit seperti Hadoop selalu mengakses Profil Instans EC2.
- Disarankan agar Anda menonaktifkan Tampilan Konsistensi EMR Amazon. S3 sangat konsisten, jadi tidak lagi diperlukan. Lihat Konsistensi kuat Amazon S3 untuk informasi lebih lanjut.
- Plugin EMRFS S3 membuat banyak panggilan STS. Direkomendasikan bahwa Anda melakukan pengujian beban pada akun pengembangan dan memantau volume panggilan STS. Anda juga disarankan untuk membuat permintaan STS untuk menaikkan batas AssumeRole layanan.

Plugin Trino

Trino (sebelumnya PrestosQL) adalah mesin query SQL yang dapat Anda gunakan untuk menjalankan kueri pada sumber data seperti HDFS, penyimpanan objek, database relasional, dan database NoSQL. Ini menghilangkan kebutuhan untuk memigrasikan data ke lokasi pusat dan memungkinkan Anda untuk menanyakan data dari kapan pun ia berada. Amazon EMR menyediakan plugin Apache Ranger untuk menyediakan kontrol akses berbutir halus untuk Trino. Plugin ini kompatibel dengan server Admin Apache Ranger versi 2.0 dan versi terbaru.

Topik

- Fitur yang didukung
- Instalasi konfigurasi layanan
- Membuat kebijakan Trino
- Pertimbangan-pertimbangan
- Batasan

Fitur yang didukung

Plugin Apache Ranger untuk Trino di Amazon EMR mendukung semua fungsionalitas mesin kueri Trino yang dilindungi oleh kontrol akses berbutir halus. Ini termasuk database, tabel, kontrol akses tingkat kolom dan pemfilteran baris dan penyembunyian data. Kebijakan Apache Ranger dapat mencakup kebijakan hibah dan tolak kebijakan untuk pengguna dan grup. Acara audit juga diserahkan ke CloudWatch log.

Instalasi konfigurasi layanan

Instalasi definisi layanan Trino mengharuskan server Admin Ranger diatur. Untuk mengatur pemisah Admin Ranger, lihat. Atur server Admin Ranger

Ikuti langkah-langkah ini untuk menginstal definisi layanan Trino.

1. SSH ke server Admin Apache Ranger.

```
ssh ec2-user@ip-xxx-xxx-xxx-xxx.ec2.internal
```

2. Copot pemasangan plugin server Presto, jika ada. Jalankan perintah berikut. Jika kesalahan ini terjadi dengan kesalahan “Layanan tidak ditemukan”, ini berarti plugin server Presto tidak diinstal di server Anda. Lanjutkan ke langkah berikutnya.

```
curl -f -u *<admin users login>*:*_<_**_password_ **_for_** _ranger admin
 user_**_>_* -X DELETE -k 'https://*<RANGER SERVER ADDRESS>*:6182/service/public/
v2/api/servicedef/name/presto'
```

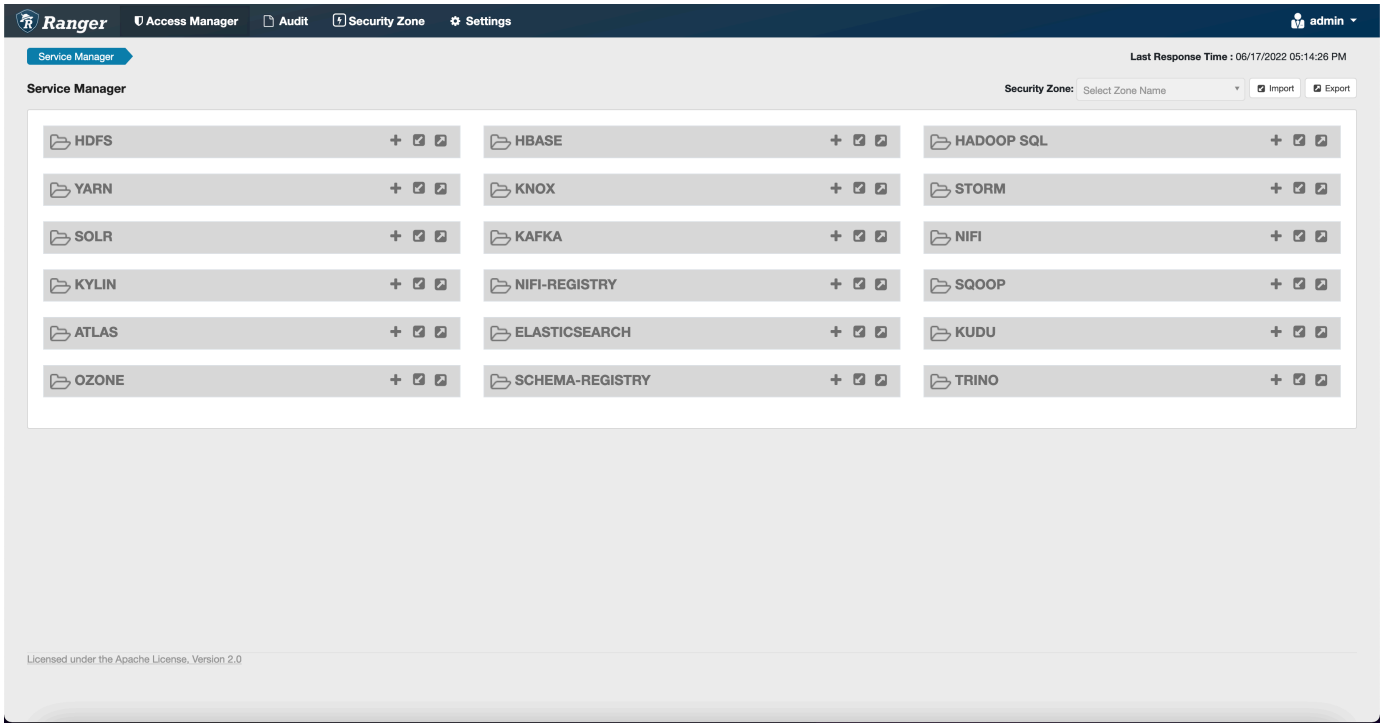
3. Unduh definisi layanan dan plugin server Admin Apache Ranger. Di direktori sementara, unduh definisi layanan. Definisi layanan ini didukung oleh versi Ranger 2.x.

```
wget https://s3.amazonaws.com/elasticmapreduce/ranger/service-definitions/
version-2.0/ranger-servicedef-amazon-emr-trino.json
```

4. Daftarkan definisi layanan Apache Trino untuk Amazon EMR.

```
curl -u *<admin users login>*:*_<_**_password_ **_for_** _ranger admin user_**_>_*
 -X POST -d @ranger-servicedef-amazon-emr-trino.json \
-H "Accept: application/json" \
-H "Content-Type: application/json" \
-k 'https://*<RANGER SERVER ADDRESS>*:6182/service/public/v2/api/servicedef'
```

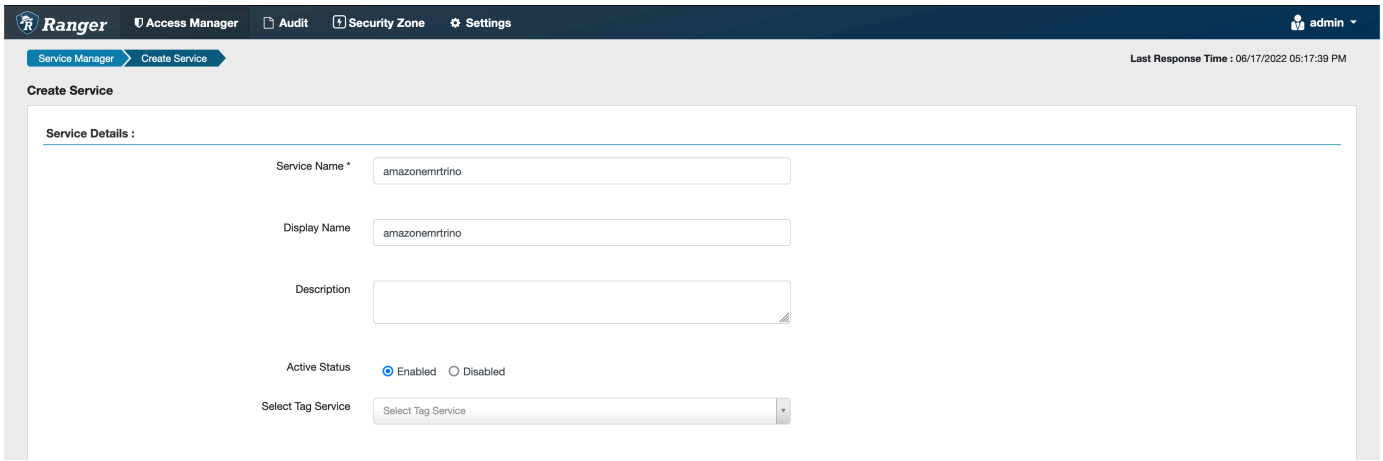
Jika perintah ini berhasil berjalan, Anda akan melihat layanan baru di UI Admin Ranger Anda dipanggilTRINO, seperti yang ditunjukkan pada gambar berikut.



5. Buat instance `TRINO` aplikasi, masukkan informasi berikut.

   Nama Layanan: Nama layanan yang akan Anda gunakan. Nilai yang direkomendasikan adalah `amazonemrtrino`. Perhatikan nama layanan ini, karena akan diperlukan saat membuat konfigurasi keamanan Amazon EMR.

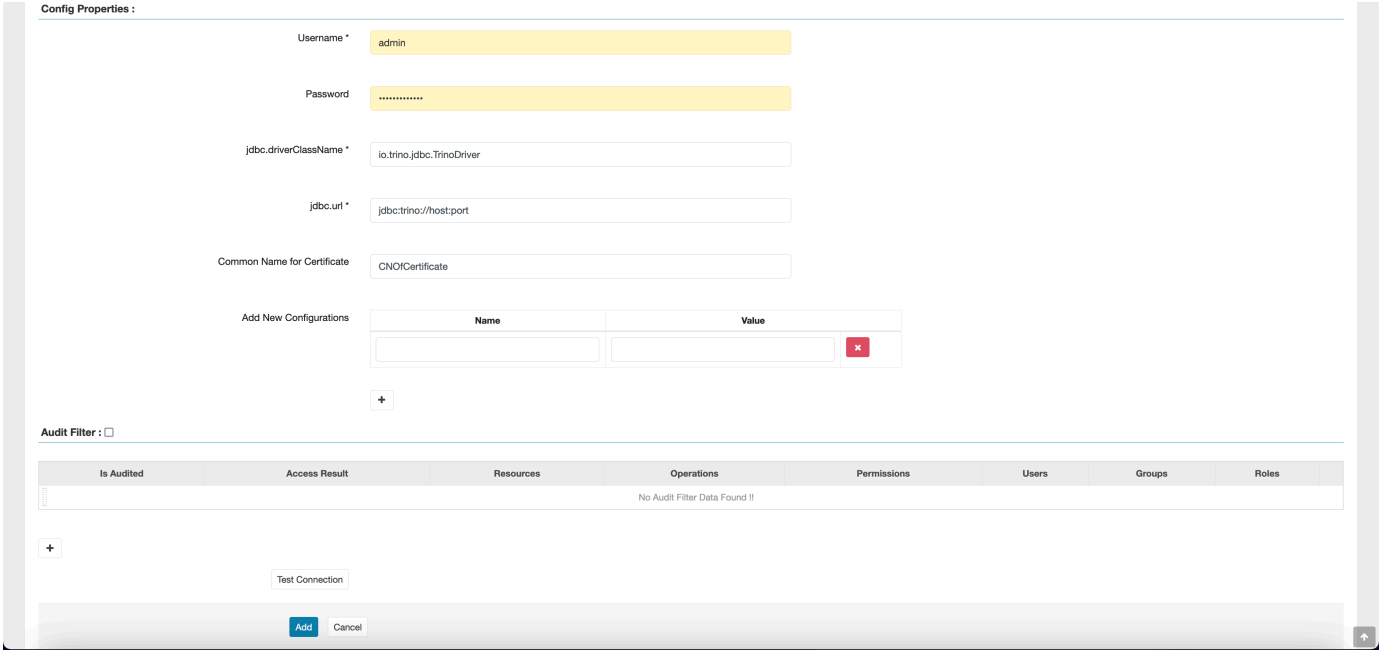
   Nama tampilan: Nama yang akan ditampilkan untuk instans ini. Nilai yang direkomendasikan adalah `amazonemrtrino`.



   jdbc.driver. ClassName: Nama kelas kelas JDBC untuk konektivitas Trino. Anda dapat menggunakan nilai default.

jdbc.url: String koneksi JDBC yang akan digunakan saat menghubungkan ke koordinator Trino.

Nama Umum Untuk Sertifikat: Bidang CN di sertifikat yang digunakan untuk connect ke server admin dari plugin klien. Nilai ini harus cocok dengan bidang CN di sertifikat TLS Anda yang dibuat untuk plugin.



Perhatikan bahwa sertifikat TLS untuk plugin ini seharusnya telah terdaftar di toko kepercayaan di server Admin Ranger. Untuk informasi selengkapnya, lihat sertifikat TLS.

Membuat kebijakan Trino

Saat Anda membuat kebijakan baru, isi kolom berikut.

Nama Kebijakan: Nama kebijakan ini.

Label Kebijakan: Label yang dapat Anda tempatkan di kebijakan ini.

Katalog: Katalog tempat kebijakan ini berlaku. Wildcard “*” mewakili semua katalog.

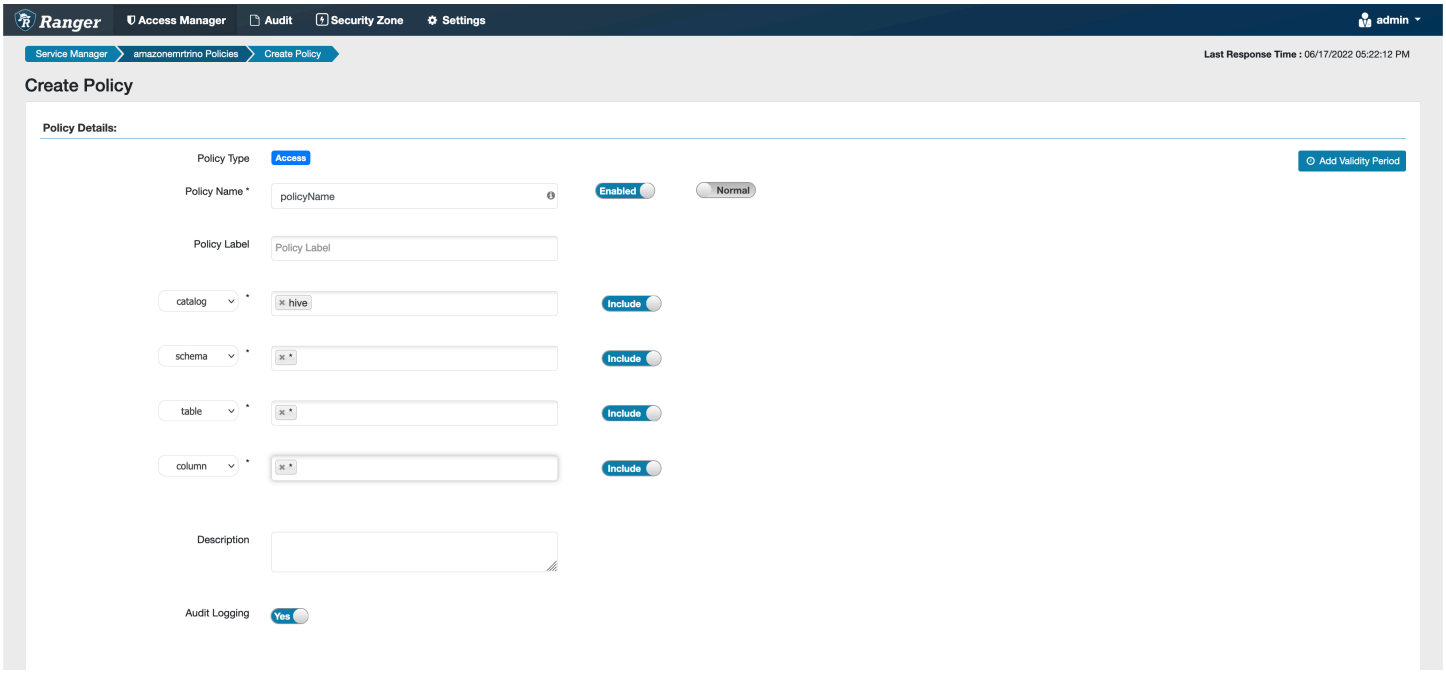
Skema: Skema yang berlaku untuk kebijakan ini. Wildcard “*” mewakili semua skema.

Tabel: Tabel yang berlaku untuk kebijakan ini. Wildcard “*” mewakili semua tabel.

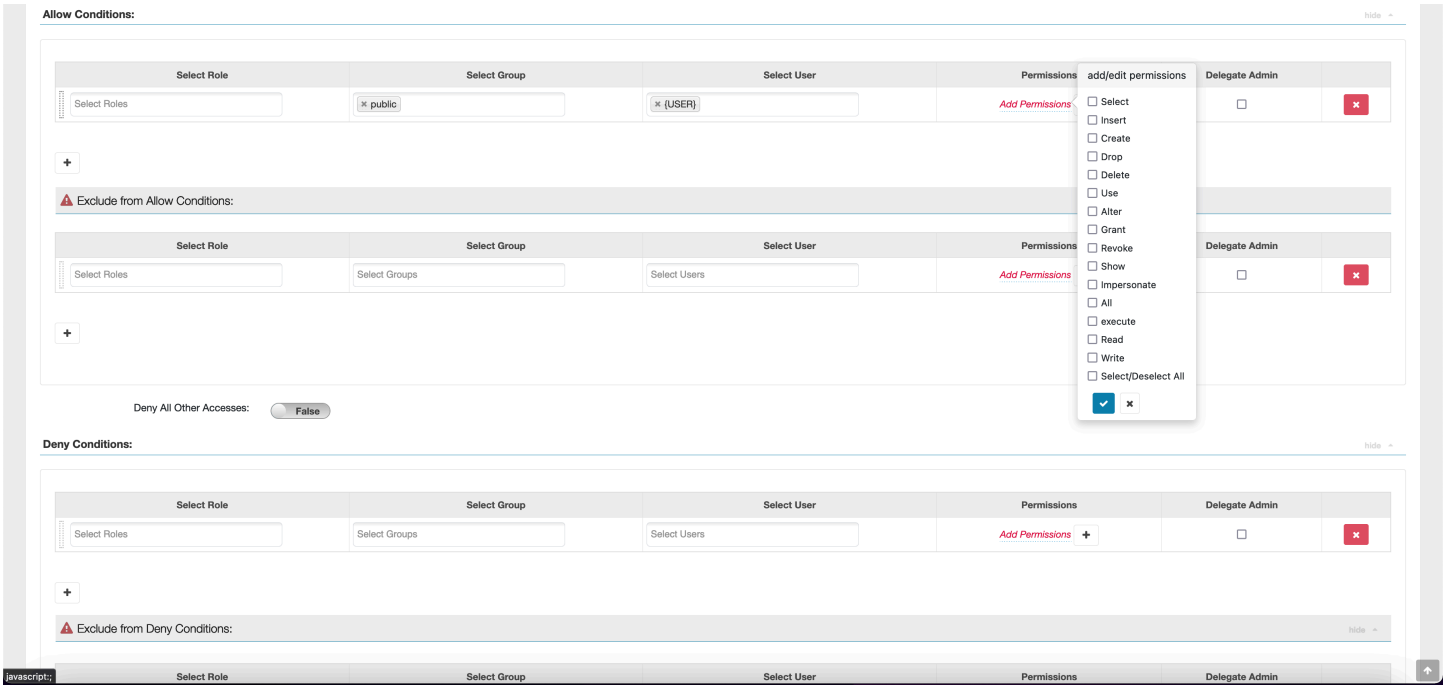
Kolom: Kolom tempat kebijakan ini berlaku. Wildcard **"*"** mewakili semua kolom.

Deskripsi: Deskripsi dari kebijakan ini.

Jenis kebijakan lain ada untuk Pengguna Trino (untuk akses peniruan identitas pengguna), Properti Sistem/Sesi Trino (untuk mengubah sistem mesin atau properti sesi), Fungsi/Prosedur (untuk memungkinkan panggilan fungsi atau prosedur), dan URL (untuk memberikan akses baca/tulis ke mesin di lokasi data).



Untuk memberikan izin kepada pengguna dan grup tertentu, masukkan pengguna dan grup. Anda juga dapat menentukan pengecualian untuk kondisi izinkan dan kondisi penolakan.



Setelah menentukan kondisi izinkan dan tolak, pilih Simpan.

Pertimbangan-pertimbangan

Saat membuat kebijakan Trino dalam Apache Ranger, ada beberapa pertimbangan penggunaan yang harus diperhatikan.

Server metadata sarang

Server metadata Hive hanya dapat diakses oleh mesin tepercaya, khususnya mesin Trino, untuk melindungi dari akses yang tidak sah. Server metadata Hive juga diakses oleh semua node di cluster. Port 9083 yang diperlukan menyediakan semua node akses ke node utama.

Otentikasi

Secara default, Trino dikonfigurasi untuk mengautentikasi menggunakan Kerberos seperti yang dikonfigurasi dalam konfigurasi keamanan Amazon EMR.

Diperlukan enkripsi dalam transit

Plugin Trino mengharuskan Anda mengaktifkan enkripsi dalam transit dalam konfigurasi keamanan EMR Amazon. Untuk mengaktifkan enkripsi, lihatEnkripsi dalam transit.

Batasan

Berikut ini adalah batasan plugin Trino saat ini:

- Server Admin Ranger tidak mendukung pelengkapan otomatis.

Penyelesaian masalah Apache Ranger

Berikut adalah beberapa masalah yang sering didiagnosis terkait penggunaan Apache Ranger.

Rekomendasi

- Uji menggunakan cluster node utama tunggal: Penyediaan cluster master node tunggal lebih cepat daripada cluster multi-node yang dapat mengurangi waktu untuk setiap iterasi pengujian.
- Atur mode pengembangan pada cluster. Ketika memulai klaster EMR Anda, atur parameter `--additional-info"` ke:

```
'{"clusterType":"development"}'
```

Parameter ini hanya dapat diatur melalui AWS CLI atau AWS SDK dan tidak tersedia melalui konsol EMR Amazon. Saat flag ini disetel, dan master gagal menyediakan, layanan EMR Amazon membuat cluster tetap hidup selama beberapa waktu sebelum menonaktifkannya. Kali ini sangat berguna untuk menyelidik berbagai log file sebelum klaster dihentikan.

Kluster EMR gagal disediakan

Ada beberapa alasan mengapa cluster EMR Amazon mungkin gagal untuk memulai. Berikut adalah beberapa cara untuk mendiagnosis masalah ini.

Periksa log penyediaan EMR

Amazon EMR menggunakan Puppet untuk menginstal dan mengkonfigurasi aplikasi pada cluster. Melihat log akan memberikan detail apakah ada kesalahan selama fase penyediaan cluster. Log dapat diakses pada klaster atau S3 jika log dikonfigurasi untuk didorong ke S3.

Log disimpan di `/var/log/provision-node/apps-phase/0/{UUID}/puppet.log` pada disk dan `s3://<LOG LOCATION>/<CLUSTER ID>/node/<EC2 INSTANCE ID>/provision-node/apps-phase/0/{UUID}/puppet.log.gz`.

Pesan Kesalahan Umum

Pesan kesalahan	Penyebab
Puppet (err): Systemd mulai gagal! emr-record-server log journalctl untuk: emr-record-server	Server Catatan EMR gagal untuk mulai. Lihat log Server Catatan EMR di bawah ini.
Puppet (err): Systemd mulai gagal! emr-record-server log journalctl untuk emrsecretagent:	Agen Rahasia EMR gagal untuk mulai. Memeriksa log Agen Rahasia di bawah ini.
/Stage [main] /Ranger_plugins: :Ranger_hive_plugin/Ranger_plugins: :prepare_two_way_tls [konfigurasikan TLS 2 arah di plugin Hive] / Exec [buat keystore dan truststore untuk plugin Ranger Hive] /return (pemberitahuan):	Sertifikat TLS privat di Secrets Manager untuk Apache Plugin Ranger sertifikat tidak dalam format yang benar atau bukan sertifikat privat. Lihat Sertifikat TLS untuk format sertifikat.

Pesan kesalahan	Penyebab
140408606197664:Error:0906D06c:PEM Rutinitas: PEM_READ_BIO:Tidak ada garis awal:PEM_LIB.c:707:Mengharapkan: KUNCI PRIBADI APA PUN	
/Stage [main] /Ranger_plugins: :Ranger_s 3_plugin/Ranger_plugins: :prepare_two_way_t ls [konfigurasikan TLS 2 arah di plugin Ranger s3] /Exec [buat keystore dan truststore untuk plugin Ranger 3] /returns (pemberitahuan): Terjadi kesalahan () saat memanggil operasi: User: arn:aws:sts: xxxxxxxxxxx:Assumed-role/ emr_ec2_ /i-xxxxxxxxxxxx tidak berwenang untuk melakukan: secretsmanager: on resource: arn:aws:secretsmanager:us-e ast-1:xxxxxxxxxx:secret: -xxxxx amazon-em r-s AccessDeniedException GetSecretValue DefaultRole GetSecretValue AdminServer	Peran profil instans EC2 tidak memiliki izin yang benar untuk mengambil sertifikat TLS dari Agen Rahasia.

Periksa SecretAgent log

Log Agen Rahasia terletak di `/emr/secretagent/log/` pada simpul EMR, atau di direktori `s3://<LOG LOCATION>/<CLUSTER ID>/node/<EC2 INSTANCE ID>/daemons/secretagent/` di S3.

Pesan Kesalahan Umum

Pesan kesalahan	Penyebab
Pengecualian di utas “main” com.amazo naws.services.securitytoken.model. AWSSecuri tyTokenServiceException: Pengguna: arn:aws:sts: :xxxxxxxxxxxx:Assumed-role/EMR_EC2_ /i-xxxxxxxxxxxxxx tidak	Pengecualian di atas berarti bahwa peran profil instans EMR EC2 tidak memiliki izin untuk mengambil peran tersebut. RangerPlu ginDataAccessRole Lihat IAM role untuk integrasi alami dengan Apache Ranger.

Pesan kesalahan	Penyebab
berwenang untuk melakukan: sts: pada sumber daya: arn:aws:iam: :xxxxxxxxxxx:peran/* * (Layanan:; Kode Status: 403; Kode Kesalahan:; DefaultRole Permintaan ID: XXXXXXXX-XXXX-XXXX-XXXX-XXXX-XXXX-XXXXXXXXXXX X; Proksi: nol) AssumeRole RangerPlu ginDataAccessRole AWSSecurityTokenService AccessDenied	
KESALAHAN qtp54617902-149: Terjadi Pengecualian Aplikasi Web javax.ws.rs. NotAllowedException: Metode HTTP 405 Tidak Diizinkan	Kesalahan ini bisa diabaikan.

Periksa Catatan Server Log (untuk SparkSQL)

<LOG LOCATION><CLUSTER ID><EC2 INSTANCE ID>Log EMR Record Server tersedia di /var/log/emr-record-server/pada simpul EMR, atau dapat ditemukan di direktori s3:////node/ /daemons//di S3. emr-record-server

Pesan Kesalahan Umum

Pesan kesalahan	Penyebab
InstanceMetadataServiceResourceFetcher:105 - [] Gagal mengambil token com.amazonaws. SdkClientException: Gagal terhubung ke titik akhir layanan	EMR SecretAgent gagal muncul atau mengalami masalah. Periksa SecretAgent log untuk kesalahan dan skrip boneka untuk menentukan apakah ada kesalahan penyediaa n.

Kueri tiba-tiba gagal

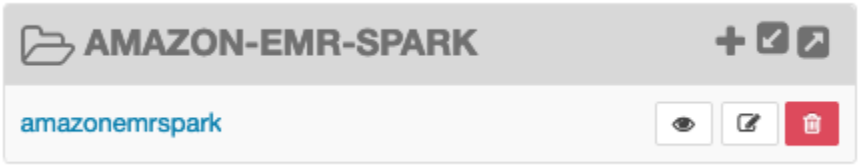
Periksa log plugin Apache Ranger (Apache Hive, RecordServer EMR, EMR, dll., SecretAgent Log)

Bagian ini umum di semua aplikasi yang terintegrasi dengan plugin Ranger, seperti Apache Hive, EMR Record Server, dan EMR. SecretAgent

Pesan Kesalahan Umum

Pesan kesalahan	Penyebab
ERROR:272 PolicyRefresher - [] (PolicyRefresherServiceName=Policy-repository): gagal menemukan layanan. Akan membersihkan cache lokal kebijakan (-1)	Pesan kesalahan ini berarti bahwa nama layanan yang Anda berikan di konfigurasi keamanan EMR tidak cocok dengan repositori kebijakan layanan di server Admin Ranger.

Jika di server Admin Ranger layanan AMAZON-EMR-SPARK Anda terlihat seperti berikut ini, maka Anda harus memasukkan **`amazonemrspark`** sebagai nama layanan.



Mengendalikan lalu lintas jaringan dengan grup keamanan

Grup keamanan bertindak sebagai firewall virtual untuk instans EC2 di klaster Anda guna mengontrol lalu lintas inbound dan keluar. Setiap grup keamanan memiliki seperangkat aturan yang mengontrol lalu lintas inbound, dan seperangkat aturan terpisah untuk mengontrol lalu lintas outbound. Untuk informasi selengkapnya, lihat Grup keamanan Amazon EC2 untuk Instans Linux di Panduan Pengguna Amazon EC2 untuk Instans Linux.

Anda menggunakan dua kelas grup keamanan dengan Amazon EMR: grup keamanan terkelola Amazon EMR dan Grup keamanan tambahan.

Setiap klaster telah mengelola grup keamanan yang terkait dengannya. Anda dapat menggunakan grup keamanan terkelola default yang dibuat Amazon EMR, atau menentukan grup keamanan terkelola khusus. Yang mana pun yang Anda pilih, Amazon EMR secara otomatis menambahkan aturan untuk grup keamanan terkelola yang klasternya perlu berkomunikasi antara instans klaster dan layanan AWS .

Grup keamanan tambahan adalah opsional. Anda dapat menentukan mereka selain grup keamanan terkelola untuk menyesuaikan akses ke instans klaster. Grup keamanan tambahan hanya berisi aturan yang Anda tetapkan. Amazon EMR tidak memodifikasi mereka.

Aturan yang Amazon EMR ciptakan di grup keamanan terkelola mengizinkan klaster untuk berkomunikasi antara komponen internal. Untuk mengizinkan pengguna dan aplikasi untuk mengakses klaster dari luar klaster, Anda dapat mengedit aturan di grup keamanan terkelola, Anda dapat membuat grup keamanan tambahan dengan aturan tambahan, atau melakukan keduanya.

> **Important**
>
> Mengedit aturan di grup keamanan terkelola mungkin memiliki konsekuensi yang tidak diinginkan. Anda mungkin secara tidak sengaja mem block lalu lintas yang diperlukan untuk klaster berfungsi dengan baik dan menyebabkan kesalahan karena simpul tidak terjangkau. Hati-hati merencanakan dan menguji konfigurasi grup keamanan sebelum implementasi.

Anda dapat menentukan grup keamanan hanya ketika Anda membuat sebuah klaster. Mereka tidak dapat ditambahkan ke klaster atau instans klaster sementara klaster berjalan, tetapi Anda dapat mengedit, menambah, dan menghapus aturan dari grup keamanan yang ada. Aturan ini berlaku segera setelah Anda menyimpannya.

Grup keamanan yang ketat secara default. Kecuali aturan ditambahkan yang mengizinkan lalu lintas, lalu lintas ditolak. Jika ada lebih dari satu aturan yang berlaku untuk lalu lintas yang sama dan sumber yang sama, aturan yang paling permisif akan berlaku. Misalnya, jika Anda memiliki aturan yang mengizinkan SSH dari alamat IP 192.0.2.12/32, dan aturan lain yang mengizinkan akses ke semua lalu lintas TCP dari kisaran 192.0.2.0/24, aturan yang mengizinkan semua lalu lintas TCP dari kisaran yang mencakup 192.0.2.12 diutamakan. Di kasus ini, klien di 192.0.2.12 mungkin memiliki akses lebih dari yang Anda inginkan.

> **Important**
>
> Berhati-hatilah saat Anda mengedit aturan grup keamanan untuk membuka port. Pastikan untuk menambahkan aturan yang hanya mengizinkan lalu lintas dari klien tepercaya dan terautentikasi untuk protokol dan port yang diperlukan untuk menjalankan beban kerja Anda.

Anda dapat mengonfigurasi akses publik blok EMR Amazon di setiap Wilayah yang Anda gunakan untuk mencegah pembuatan klaster jika aturan mengizinkan akses publik pada port apa pun yang

tidak Anda tambahkan ke daftar pengecualian. Untuk akun AWS yang dibuat setelah Juli 2019, akses publik blok Amazon EMR aktif secara default. Untuk akun AWS yang membuat klaster sebelum Juli 2019, akses publik blok Amazon EMR nonaktif secara default. Untuk informasi selengkapnya, lihat Menggunakan Akses publik blok Amazon EMR.

Topik

> **Note**
>
> Amazon EMR bertujuan untuk menggunakan alternatif inklusif untuk istilah industri yang berpotensi menyinggung atau non-inklusif seperti “master” dan “slave”. Kami telah beralih ke terminologi baru untuk menumbuhkan pengalaman yang lebih inklusif dan untuk memfasilitasi pemahaman Anda tentang komponen layanan.
> Kami sekarang mendeskripsikan “node” sebagai instance, dan kami menjelaskan jenis instans EMR Amazon sebagai instance primer, inti, dan tugas. Selama transisi, Anda mungkin masih menemukan referensi lama ke istilah yang sudah ketinggalan zaman, seperti yang berkaitan dengan grup keamanan untuk Amazon EMR.

Bekerja dengan grup keamanan terkelola Amazon EMR

> **Note**
>
> Amazon EMR bertujuan untuk menggunakan alternatif inklusif untuk istilah industri yang berpotensi menyinggung atau non-inklusif seperti “master” dan “slave”. Kami telah beralih ke terminologi baru untuk menumbuhkan pengalaman yang lebih inklusif dan untuk memfasilitasi pemahaman Anda tentang komponen layanan.
> Kami sekarang mendeskripsikan “node” sebagai instance, dan kami menjelaskan jenis instans EMR Amazon sebagai instance primer, inti, dan tugas. Selama transisi, Anda

mungkin masih menemukan referensi lama ke istilah yang sudah ketinggalan zaman, seperti yang berkaitan dengan grup keamanan untuk Amazon EMR.

Grup keamanan terkelola yang berbeda dikaitkan dengan instance utama dan dengan instance inti dan tugas dalam sebuah cluster. Grup keamanan terkelola tambahan untuk akses layanan diperlukan ketika Anda membuat sebuah klaster di subnet privat. Untuk informasi selengkapnya tentang peran grup keamanan terkelola sehubungan dengan konfigurasi jaringan Anda, lihat Opsi Amazon VPC.

Ketika Anda menetapkan grup keamanan terkelola untuk sebuah klaster, Anda harus menggunakan tipe grup keamanan yang sama, default atau kustom, untuk semua grup keamanan terkelola. Misalnya, Anda tidak dapat menentukan grup keamanan khusus untuk instance utama, lalu tidak menentukan grup keamanan khusus untuk instance inti dan tugas.

Jika Anda menggunakan grup keamanan terkelola default, Anda tidak perlu menentukannya saat membuat klaster. Amazon EMR secara otomatis menggunakan default. Selain itu, jika default tidak belum ada di klaster VPC, Amazon EMR akan membuatnya. Amazon EMR juga menciptakan mereka jika Anda secara eksplisit menentukan mereka dan mereka belum ada.

Anda dapat mengedit aturan di grup keamanan terkelola setelah klaster dibuat. Ketika Anda membuat sebuah klaster baru, Amazon EMR memeriksa aturan di grup keamanan terkelola yang Anda tentukan, dan membuat aturan inbound yang hilang bahwa kebutuhan klaster baru selain aturan yang mungkin telah ditambahkan sebelumnya. Kecuali dinyatakan lain secara khusus, setiap aturan untuk grup keamanan yang dikelola Amazon EMR default juga ditambahkan ke grup keamanan terkelola Amazon EMR khusus yang Anda tentukan.

Grup keamanan terkelola default adalah sebagai berikut:

- ElasticMapReduce-primer

  Untuk aturan di grup keamanan ini, lihat Grup keamanan yang dikelola Amazon EMR untuk instans utama (subnet publik).
- ElasticMapReduce-inti

  Untuk aturan di grup keamanan ini, lihat Grup keamanan terkelola Amazon EMR untuk instans inti dan instans tugas (subnet publik).
- ElasticMapReduce-Primer-Pribadi

Untuk aturan di grup keamanan ini, lihat Grup keamanan yang dikelola Amazon EMR untuk instans utama (subnet pribadi).

- ElasticMapReduce-Inti-Pribadi

  Untuk aturan di grup keamanan ini, lihat Grup keamanan terkelola Amazon EMR untuk instans inti dan instans tugas (subnet privat).

- ElasticMapReduce-ServiceAccess

  Untuk aturan di grup keamanan ini, lihat Grup keamanan terkelola Amazon EMR untuk akses layanan (subnet privat).

Grup keamanan yang dikelola Amazon EMR untuk instans utama (subnet publik)

Grup keamanan terkelola default untuk instance utama dalam subnet publik memiliki Nama Grup ElasticMapReduce -primary. Aplikasi ini memiliki aturan berikut: Jika Anda menentukan grup keamanan terkelola kustom, Amazon EMR menambahkan semua aturan yang sama ke grup keamanan kustom Anda.

Tipe	Protok	Rentar Port	Sumber	Detail
Aturan-aturan ke dalam				
Semua ICMP-IPv4	Semua	T/A	ID Grup grup keamanan terkelola untuk contoh utama. Dengan kata lain, grup keamanan yang sama di mana aturan muncul.	Aturan refleksif ini mengizinkan lalu lintas inbound dari setiap instans yang terkait dengan grup keamanan tertentu. Menggunakan `ElasticMapReduce-primary` default untuk beberapa klaster mengizinkan simpul inti dan simpul tugas klaster tersebut untuk berkomunikasi dengan satu sama lain melalui ICMP atau port TCP atau UDP. Tentukan grup keamanan terkelola kustom untuk membatasi akses lintas-klaster.
Semua TCP	TCP	Semua		
Semua UDP	UDP	Semua		

Tipe	Protok	Rentar Port	Sumber	Detail
Semua ICMP-IPV4	Semua	N/A	ID Grup dari grup keamanan terkelola yang ditentukan untuk simpul inti dan simpul tugas.	Aturan-aturan ini mengizinkan semua lalu lintas ICMP inbound dan lalu lintas di atas setiap port TCP atau UDP dari setiap instans inti dan instans tugas yang terkait dengan grup keamanan tertentu, bahkan jika instans di klaster yang berbeda.
Semua TCP	TCP	Semua		
Semua UDP	UDP	Semua		
Kustom	TCP	8443	Berbagai rentang alamat IP Amazon	Aturan-aturan ini memungkinkan pengelola cluster untuk berkomunikasi dengan node utama.

Untuk memberikan akses SSH sumber tepercaya ke grup keamanan utama dengan konsol lama

Untuk mengedit grup keamanan, Anda harus memiliki izin untuk mengelola grup keamanan untuk VPC tempat klaster berada. Untuk informasi selengkapnya, lihat Mengubah Izin untuk pengguna dan Kebijakan Contoh yang memungkinkan mengelola grup keamanan EC2 di Panduan Pengguna IAM.

1. Arahkan ke konsol EMR Amazon baru dan pilih Beralih ke konsol lama dari navigasi samping. Untuk informasi selengkapnya tentang apa yang diharapkan saat beralih ke konsol lama, lihat Menggunakan konsol lama.
2. Pilih Klaster. Pilih Nama cluster yang ingin Anda modifikasi.
3. Pilih tautan Grup keamanan untuk Master di bawah Keamanan dan akses.
4. Pilih ElasticMapReduce-master dari daftar.
5. Pilih tab Aturan masuk dan kemudian Edit aturan masuk.
6. Memeriksa aturan masuk yang mengizinkan akses publik dengan pengaturan berikut. Jika ada, pilih Hapus untuk menghapusnya.

   - Tipe

     SSH
   - Pelabuhan

22

- Sumber

  Kustom 0.0.0.0/0

> **Warning**
>
> Sebelum Desember 2020, grup keamanan ElasticMapReduce -master memiliki aturan yang telah dikonfigurasi sebelumnya untuk mengizinkan lalu lintas masuk di Port 22 dari semua sumber. Aturan ini dibuat untuk menyederhanakan koneksi SSH awal ke node utama. Kami sangat menyarankan agar Anda menghapus aturan masuk ini dan membatasi lalu lintas ke sumber tepercaya.

7. Gulir ke bagian bawah daftar aturan dan pilih Tambahkan Aturan.
8. Untuk Jenis, pilih SSH.

   Memilih SSH secara otomatis memasuki TCP untuk Protokol dan 22 untuk Rentang Port.
9. Untuk sumber, pilih IP Saya untuk secara otomatis menambahkan alamat IP Anda sebagai alamat sumber. Anda juga dapat menambahkan berbagai alamat IP klien tepercaya kustom, atau membuat aturan tambahan untuk klien lain. Banyak lingkungan jaringan mengalokasikan alamat IP secara dinamis, jadi Anda mungkin perlu memperbarui alamat IP Anda untuk klien tepercaya di masa mendatang.
10. Pilih Simpan.
11. Secara opsional, pilih ElasticMapReduce-slave dari daftar dan ulangi langkah-langkah di atas untuk memungkinkan klien SSH mengakses node inti dan tugas.

Grup keamanan terkelola Amazon EMR untuk instans inti dan instans tugas (subnet publik)

Grup keamanan terkelola default untuk instance inti dan tugas di subnet publik memiliki Nama Grup - core. ElasticMapReduce Grup keamanan terkelola default memiliki aturan berikut, dan Amazon EMR menambahkan aturan yang sama jika Anda menentukan grup keamanan terkelola kustom.

Tipe	Protok	Rentar Port	Sumber	Detail
Aturan-aturan ke dalam				
Semua ICMP-IPV4	Semua	N/A	ID Grup dari grup keamanan terkelola untuk instans inti dan instans tugas. Dengan kata lain, grup keamanan yang sama di mana aturan muncul.	Aturan refleksif ini mengizinkan lalu lintas inbound dari setiap instans yang terkait dengan grup keamanan tertentu. Menggunak an `ElasticMapReduce-core` default untuk beberapa klaster mengizinkan instans inti dan instans tugas dari klaster tersebut untuk berkomunikasi satu sama lain melalui ICMP atau port TCP atau UDP. Tentukan grup keamanan terkelola kustom untuk membatasi akses lintas-klaster.
Semua TCP	TCP	Semua		
Semua UDP	UDP	Semua		
Semua ICMP-IPV4	Semua	T/A	ID Grup grup keamanan terkelola untuk contoh utama.	Aturan ini memungkinkan semua lalu lintas ICMP masuk dan lalu lintas melalui port TCP atau UDP apa pun dari instance utama apa pun yang terkait dengan grup keamanan yang ditentukan, bahkan jika instance berada dalam kelompok yang berbeda.
Semua TCP	TCP	Semua		
Semua UDP	UDP	Semua		

Grup keamanan yang dikelola Amazon EMR untuk instans utama (subnet pribadi)

Grup keamanan terkelola default untuk instance utama dalam subnet pribadi memiliki Nama Grup ElasticMapReduce -Primary-Private. Grup keamanan terkelola default memiliki aturan berikut, dan Amazon EMR menambahkan aturan yang sama jika Anda menentukan grup keamanan terkelola kustom.

Tipe	Protok	Rentar Port	Sumber	Detail
Aturan-aturan ke dalam				

Tipe	Protok	Rentar Port	Sumber	Detail
Semua ICMP-IPv4	Semua	T/A	ID Grup grup keamanan terkelola untuk contoh utama. Dengan kata lain, grup keamanan yang sama di mana aturan muncul.	Aturan refleksif ini mengizinkan lalu lintas inbound dari setiap instans yang terkait dengan grup keamanan tertentu dan dapat dicapai dari dalam subnet privat. Menggunakan `ElasticMapReduce-Primary-Private` default untuk beberapa klaster mengizinkan simpul inti dan simpul tugas klaster tersebut untuk berkomunikasi dengan satu sama lain melalui ICMP atau port TCP atau UDP. Tentukan grup keamanan terkelola kustom untuk membatasi akses lintas-klaster.
Semua TCP	TCP	Semua		
Semua UDP	UDP	Semua		
Semua ICMP-IPV4	Semua	N/A	ID Grup dari grup keamanan terkelola untuk simpul inti dan simpul tugas.	Aturan-aturan ini mengizinkan semua lalu lintas ICMP inbound dan lalu lintas di atas port TCP atau UDP dari setiap instans inti dan instans tugas yang terkait dengan grup keamanan tertentu dan terjangkau dari di subnet privat, bahkan jika instans di klaster yang berbeda.
Semua TCP	TCP	Semua		
Semua UDP	UDP	Semua		
HTTPS (8443)	TCP	8443	ID Grup dari grup keamanan terkelola untuk akses layanan di subnet privat.	Aturan ini memungkinkan pengelola cluster untuk berkomunikasi dengan node utama.
Aturan-aturan ke luar				
Semua lalu lintas	Semua	Semua	0.0.0.0/0	Menyediakan akses outbound ke internet.

Tipe	Protok	Rentar Port	Sumber	Detail
TCP kustom	TCP	9443	ID Grup dari grup keamanan terkelola untuk akses layanan di subnet privat.	Jika aturan keluar default “Semua lalu lintas” di atas dihapus, aturan ini adalah persyaratan minimum untuk Amazon EMR 5.30.0 dan yang lebih baru. **Note** Amazon EMR tidak menambahkan aturan ini saat Anda menggunakan grup keamanan terkelola khusus.
TCP Kustom	TCP	80 (http) atau 443 (https)	ID Grup dari grup keamanan terkelola untuk akses layanan di subnet privat.	Jika aturan keluar default “Semua lalu lintas” di atas dihapus, aturan ini adalah persyaratan minimum untuk Amazon EMR 5.30.0 dan yang lebih baru untuk terhubung ke Amazon S3 melalui https. **Note** Amazon EMR tidak menambahkan aturan ini saat Anda menggunakan grup keamanan terkelola khusus.

Grup keamanan terkelola Amazon EMR untuk instans inti dan instans tugas (subnet privat)

Grup keamanan terkelola default untuk instance inti dan tugas di subnet pribadi memiliki Nama Grup -Core-Private. ElasticMapReduce Grup keamanan terkelola default memiliki aturan berikut, dan Amazon EMR menambahkan aturan yang sama jika Anda menentukan grup keamanan terkelola kustom.

Tipe	Protok	Rentar Port	Sumber	Detail
Aturan-aturan ke dalam				
Semua ICMP-IPV4	Semua	N/A	ID Grup dari grup keamanan terkelola untuk instans inti dan instans tugas. Dengan kata lain, grup keamanan yang sama di mana aturan muncul.	Aturan refleksif ini mengizinkan lalu lintas inbound dari setiap instans yang terkait dengan grup keamanan tertentu. Menggunak an `ElasticMapReduce-core` default untuk beberapa klaster mengizinkan instans inti dan instans tugas dari klaster tersebut untuk berkomunikasi satu sama lain melalui ICMP atau port TCP atau UDP. Tentukan grup keamanan terkelola kustom untuk membatasi akses lintas-klaster.
Semua TCP	TCP	Semua		
Semua UDP	UDP	Semua		
Semua ICMP-IPV4	Semua	T/A	ID Grup grup keamanan terkelola untuk contoh utama.	Aturan ini memungkinkan semua lalu lintas ICMP masuk dan lalu lintas melalui port TCP atau UDP apa pun dari instance utama apa pun yang terkait dengan grup keamanan yang ditentukan, bahkan jika instance berada dalam kelompok yang berbeda.
Semua TCP	TCP	Semua		
Semua UDP	UDP	Semua		
HTTPS (8443)	TCP	8443	ID Grup dari grup keamanan terkelola untuk akses layanan di subnet privat.	Aturan ini mengizinkan manajer klaster untuk berkomunikasi dengan simpul inti dan simpul tugas.
Aturan-aturan ke luar				
Semua lalu lintas	Semua	Semua	0.0.0.0/0	Lihat Mengedit aturan outbound di bawah ini.
TCP Kustom	TCP	80 (http)	ID Grup dari grup keamanan	Jika aturan keluar default “Semua lalu lintas” di atas dihapus, aturan ini adalah persyarat

Tipe	Protok	Rentar Port	Sumber	Detail
		atau 443 (https)	terkelola untuk akses layanan di subnet privat.	an minimum untuk Amazon EMR 5.30.0 dan yang lebih baru untuk terhubung ke Amazon S3 melalui https. **Note** Amazon EMR tidak menambahkan aturan ini saat Anda menggunakan grup keamanan terkelola khusus.

Mengedit aturan outbound

Secara default, Amazon EMR menciptakan grup keamanan ini dengan aturan outbound yang mengizinkan semua lalu lintas keluar pada semua protokol dan port. Mengizinkan semua lalu lintas keluar dipilih karena berbagai Amazon EMR dan aplikasi pelanggan yang dapat berjalan di klaster Amazon EMR mungkin memerlukan aturan outbound yang berbeda. Amazon EMR tidak dapat mengantisipasi pengaturan khusus ini saat membuat grup keamanan default. Anda dapat cakupan jalan keluar di grup keamanan Anda untuk menyertakan hanya aturan-aturan yang sesuai dengan kasus penggunaan dan kebijakan keamanan Anda. Minimal, grup keamanan ini memerlukan aturan outbound berikut, tetapi beberapa aplikasi mungkin memerlukan jalan keluar tambahan.

Tipe	Protok	Rentar Port	Tujuan	Detail
Semua TCP	TCP	Semua	pl-*xxxxxxxx*	Daftar prefiks Amazon S3 terkelola `com.amazonaws.` *MyRegion*`.s3`.
Semua lalu lintas	Semua	Semua	sg-*xxxxxxxxx xxxxxxxx*	ID dari `ElasticMapReduce-Core-Private` grup keamanan.
Semua lalu lintas	Semua	Semua	sg-*xxxxxxxxx xxxxxxxx*	ID dari `ElasticMapReduce-Primary-Private` grup keamanan.

Tipe	Protok	Rentar Port	Tujuan	Detail
TCP kustom	TCP	9443	sg-*xxxxxxxxx xxxxxxxx*	ID dari `ElasticMapReduce-S erviceAccess` grup keamanan.

Grup keamanan terkelola Amazon EMR untuk akses layanan (subnet privat)

Grup keamanan terkelola default untuk akses layanan di subnet pribadi memiliki Nama Grup ElasticMapReduce - ServiceAccess. Memiliki aturan inbound, dan aturan outbound yang mengizinkan lalu lintas melalui HTTPS (port 8443, port 9443) untuk grup keamanan terkelola lainnya di subnet privat. Aturan-aturan ini memungkinkan pengelola cluster untuk berkomunikasi dengan node utama dan dengan node inti dan tugas. Aturan yang sama diperlukan jika Anda menggunakan grup keamanan kustom.

Tipe	Protok	Rentar Port	Sumber	Detail
Aturan masuk Diperlukan untuk klaster EMR Amazon dengan rilis EMR Amazon 5.30.0 dan yang lebih baru.				
TCP Kustom	TCP	9443	ID Grup grup keamanan terkelola untuk contoh utama.	Aturan ini memungkinkan komunikasi antara grup keamanan instans utama ke grup keamanan akses layanan.
Aturan keluar Diperlukan untuk semua kluster EMR Amazon				
TCP Kustom	TCP	8443	ID Grup grup keamanan terkelola untuk contoh utama.	Aturan-aturan ini memungkinkan pengelola cluster untuk berkomunikasi dengan node utama dan dengan node inti dan tugas.
TCP Kustom	TCP	8443	ID Grup dari grup keamanan terkelola untuk instans inti dan instans tugas.	Aturan-aturan ini memungkinkan pengelola cluster untuk berkomunikasi dengan node utama dan dengan node inti dan tugas.

Bekerja dengan grup keamanan tambahan

Baik Anda menggunakan grup keamanan terkelola default atau menentukan grup keamanan terkelola kustom, Anda dapat menggunakan grup keamanan tambahan. Grup keamanan tambahan memberi Anda fleksibilitas untuk menyesuaikan akses antara grup yang berbeda dan dari klien, sumber daya, dan aplikasi eksternal.

Misalnya, pertimbangkan alur perencanaan berikut ini: Anda memiliki beberapa cluster yang Anda butuhkan untuk berkomunikasi satu sama lain, tetapi Anda ingin mengizinkan akses SSH masuk ke instance utama hanya untuk subset tertentu dari cluster. Untuk melakukannya, Anda dapat menggunakan set grup keamanan terkelola yang sama untuk klaster. Anda kemudian membuat grup keamanan tambahan yang memungkinkan akses SSH masuk dari klien tepercaya, dan menentukan grup keamanan tambahan untuk instance utama ke setiap cluster dalam subset.

Anda dapat menerapkan hingga empat grup keamanan tambahan untuk instance utama, empat untuk instance inti dan tugas, dan empat untuk akses layanan (dalam subnet pribadi). Jika perlu, Anda dapat menentukan grup keamanan tambahan yang sama untuk instance utama, instance inti dan tugas, dan akses layanan. Jumlah maksimum grup keamanan dan aturan di akun Anda tunduk pada batas akun. Untuk informasi selengkapnya, lihat Batas grup keamanan di Panduan Pengguna Amazon VPC.

Menentukan grup keamanan terkelola Amazon EMR dan grup keamanan tambahan

Anda dapat menentukan grup keamanan menggunakan AWS Management ConsoleAWS CLI, API EMR Amazon, atau Amazon. Jika Anda tidak menentukan grup keamanan, Amazon EMR akan menggunakan grup keamanan default. Menentukan grup keamanan tambahan adalah opsional. Anda dapat menetapkan grup keamanan tambahan untuk instance utama, instance inti dan tugas, dan akses layanan (hanya subnet pribadi).

New console

> **Note**
>
> Kami telah mendesain ulang konsol EMR Amazon agar lebih mudah digunakan. Lihat Apa yang baru dengan konsol? untuk mempelajari tentang perbedaan antara pengalaman konsol lama dan baru.

Untuk menentukan grup keamanan dengan konsol baru

1. Masuk keAWS Management Console, dan buka konsol EMR Amazon di https://console.aws.amazon.com/emr.
2. Di bawah EMR pada EC2 di panel navigasi kiri, pilih Clusters, lalu pilih Create cluster.
3. Di bawah Jaringan, pilih panah di sebelah grup keamanan EC2 (firewall) untuk memperluas bagian ini. Di bawah Node utama dan inti dan node tugas, grup keamanan terkelola Amazon EMR default dipilih secara default. Jika Anda menggunakan subnet pribadi, Anda juga memiliki opsi untuk memilih grup keamanan untuk akses Layanan.
4. Untuk mengubah grup keamanan terkelola Amazon EMR Anda, gunakan menu tarik-turun Pilih grup keamanan untuk memilih opsi lain dari daftar opsi grup keamanan yang dikelola Amazon EMR. Anda memiliki satu grup keamanan terkelola EMR Amazon untuk node Primer dan Core dan node tugas.
5. Untuk menambahkan grup keamanan khusus, gunakan menu tarik-turun Pilih grup keamanan yang sama untuk memilih hingga empat grup keamanan kustom dari daftar opsi grup keamanan kustom. Anda dapat memiliki hingga empat grup keamanan khusus untuk node Primer dan Core dan node tugas.
6. Pilih opsi lain yang berlaku untuk cluster Anda.
7. Untuk meluncurkan klaster Anda, pilih Buat klaster.

Old console

Untuk menentukan grup keamanan dengan konsol lama

1. Arahkan ke konsol EMR Amazon baru dan pilih Beralih ke konsol lama dari navigasi samping. Untuk informasi selengkapnya tentang apa yang diharapkan saat beralih ke konsol lama, lihat Menggunakan konsol lama.
2. Memilih Buat klaster, Pergi ke opsi lanjutan.
3. Memilih opsi untuk klaster hingga mencapaiLangkah 4: Keamanan.
4. Memilih Grup Keamanan EC2 untuk memperluas bagian.

   Di bawah Grup keamanan terkelola EMR, grup keamanan terkelola default dipilih secara default. Jika default tidak ada di VPC untuk Utama, Inti & Tugas, atau Akses Layanan (hanya subnet privat), Buat muncul sebelum nama grup keamanan terkait.

5. Jika Anda menggunakan grup keamanan terkelola kustom, Anda harus memilih mereka dari daftar Grup keamanan terkelola EMR.

   Jika Anda membuat pilihan pada grup keamanan terkelola kustom, pesan akan memberi tahu Anda untuk memilih grup keamanan kustom untuk instans lainnya. Anda dapat menggunakan hanya grup keamanan terkelola kustom atau default untuk sebuah klaster.
6. Opsional, di bawah Grup keamanan tambahan, Anda perlu membuat pilihan pada ikon pensil, memilih hingga empat grup keamanan dari daftar, lalu memilih Menugaskan grup keamanan. Ulangi untuk masing-masing Utama, Inti & Tugas, dan Akses Layanan seperti yang diinginkan.
7. Memilih Buat Klaster.

Menentukan kelompok keamanan dengan AWS CLI

Untuk menentukan grup keamanan menggunakan AWS CLI Anda menggunakan perintah `create-cluster` dengan parameter opsi `--ec2-attributes` berikut:

Parameter	Deskripsi
`EmrManagedPrimarySecurityGroup`	Gunakan parameter ini untuk menentuka n grup keamanan terkelola kustom untuk instance utama. Jika parameter ini ditentuka n, `EmrManagedCoreSecurityGroup` juga harus ditentukan. Untuk klaster di subnet privat, `ServiceAccessSecurityGroup` juga harus ditentukan.
`EmrManagedCoreSecurityGroup`	Gunakan parameter ini untuk menentukan grup keamanan terkelola kustom untuk instans inti dan instans tugas. Jika parameter ini ditentuka n, `EmrManagedPrimarySecurityGroup` juga harus ditentukan. Untuk klaster di subnet privat, `ServiceAccessSecurityGroup` juga harus ditentukan.

Parameter	Deskripsi
`ServiceAccessSecurityGroup`	Gunakan parameter ini untuk menentukan grup keamanan terkelola kustom untuk akses layanan, yang hanya berlaku untuk klaster di subnet privat. Grup keamanan yang Anda tentukan sebagai `ServiceAccessSecurityGroup` tidak boleh digunakan untuk tujuan lain dan juga harus disediakan untuk Amazon EMR. Jika parameter ini ditentukan, `EmrManagedPrimarySecurityGroup` juga harus ditentukan.
`AdditionalPrimarySecurityGroups`	Gunakan parameter ini untuk menentukan hingga empat grup keamanan tambahan untuk contoh utama.
`AdditionalCoreSecurityGroups`	Gunakan parameter ini untuk menentukan hingga empat grup keamanan tambahan untuk instans inti dan instans tugas.

Example — tentukan grup keamanan terkelola Amazon EMR kustom dan grup keamanan tambahan

Contoh berikut menentukan grup keamanan terkelola Amazon EMR khusus untuk klaster di subnet pribadi, beberapa grup keamanan tambahan untuk instans utama, dan satu grup keamanan tambahan untuk instance inti dan tugas.

> **Note**
>
> Karakter lanjutan baris Linux (\) disertakan agar mudah dibaca Karakter ini bisa dihapus atau digunakan dalam perintah Linux. Untuk Windows, hapus atau ganti dengan caret (^).

```
aws emr create-cluster --name "ClusterCustomManagedAndAdditionalSGs" \
--release-label emr-emr-7.0.0 --applications Name=Hue Name=Hive \
```

```
Name=Pig --use-default-roles --ec2-attributes \
SubnetIds=subnet-xxxxxxxxxxxx,KeyName=myKey,\
ServiceAccessSecurityGroup=sg-xxxxxxxxxxxx,\
EmrManagedPrimarySecurityGroup=sg-xxxxxxxxxxxx,\
EmrManagedCoreSecurityGroup=sg-xxxxxxxxxxx,\
AdditionalPrimarySecurityGroups=['sg-xxxxxxxxxxx',\
'sg-xxxxxxxxxxx','sg-xxxxxxxxxx'],\
AdditionalCoreSecurityGroups=sg-xxxxxxxxxxx \
--instance-type m5.xlarge
```

Untuk informasi selengkapnya, lihat create-cluster di AWS CLI Referensi.

Menentukan grup-grup keamanan EC2 untuk EMR Notebooks

Saat Anda membuat buku catatan EMR, dua grup keamanan digunakan untuk mengontrol lalu lintas jaringan antara notebook EMR dan klaster EMR Amazon saat Anda menggunakan editor notebook. Grup keamanan default memiliki aturan minimal yang mengizinkan hanya lalu lintas jaringan antara layanan EMR Notebooks dan klaster yang terlampir pada notebook.

Sebuah notebook EMR menggunakan Apache Livy untuk berkomunikasi dengan cluster melalui proxy melalui TCP Port 18888. Saat membuat grup keamanan khusus dengan aturan yang disesuaikan dengan lingkungan, Anda dapat membatasi lalu lintas jaringan sehingga hanya sebagian buku catatan yang dapat menjalankan kode dalam editor notebook pada cluster tertentu. Cluster menggunakan keamanan kustom Anda selain grup keamanan default untuk cluster. Untuk informasi selengkapnya, lihat Kendalikan lalu lintas jaringan dengan grup keamanan di Panduan Pengelolaan Amazon EMR dan Menentukan grup-grup keamanan EC2 untuk EMR Notebooks.

Grup keamanan EC2 default untuk instans utama

Grup keamanan EC2 default untuk instance utama dikaitkan dengan instance utama selain grup keamanan klaster untuk instance utama.

Nama Grup: ElasticMapReduceEditors-Livy

Aturan

- Inbound

  Mengizinkan TCP Port 18888 dari sumber daya apa pun di grup keamanan default EC2 untuk EMR Notebooks
- Outbound

Tidak ada

Default grup keamanan EC2 untuk EMR Notebooks

Grup keamanan EC2 default untuk EMR Notebooks dikaitkan dengan editor notebook untuk EMR Notebooks yang ditugaskan.

Nama Grup: ElasticMapReduceEditors-Editor

Aturan

- Inbound

  Tidak ada
- Outbound

  Mengizinkan TCP Port 18888 untuk sumber daya apa pun di grup keamanan EC2 default untuk EMR Notebooks.

Grup keamanan EC2 kustom untuk EMR Notebooks ketika menghubungkan Notebook dengan repositori Git

Untuk menautkan repositori Git ke buku catatan Anda, grup keamanan untuk buku catatan EMR harus menyertakan aturan keluar sehingga buku catatan dapat merutekan lalu lintas ke internet. Dianjurkan agar Anda membuat grup keamanan baru untuk tujuan ini. Memperbarui grup keamanan ElasticMapReduceEditors-Editor default dapat memberikan aturan keluar yang sama ke buku catatan lain yang dilampirkan ke grup keamanan ini.

Aturan

- Inbound

  Tidak ada
- Outbound

  Izinkan notebook untuk merutekan lalu lintas ke internet melalui cluster, seperti yang ditunjukkan contoh berikut. Nilai 0.0.0.0/0 digunakan untuk tujuan contoh. Anda dapat memodifikasi aturan ini untuk menentukan alamat IP untuk repositori berbasis Git Anda.

Tipe	Protokol	Rentang Port	Tujuan
Aturan TCP kustom	TCP	18888	SG-
HTTPS	TCP	443	0.0.0.0/0

Menggunakan Akses publik blok Amazon EMR

Amazon EMR memblokir akses publik (BPA) mencegah Anda meluncurkan cluster di subnet publik jika cluster memiliki konfigurasi keamanan yang memungkinkan lalu lintas masuk dari alamat IP publik pada port.

> **Important**
>
> Blokir akses publik diaktifkan secara default. Untuk meningkatkan perlindungan akun, kami sarankan Anda tetap mengaktifkannya.

Memahami memblokir akses publik

Anda dapat menggunakan konfigurasi tingkat akun akses publik blokir untuk mengelola akses jaringan publik secara terpusat ke kluster Amazon EMR.

Saat pengguna dari Anda Akun AWS meluncurkan kluster, Amazon EMR memeriksa aturan port di grup keamanan untuk klaster dan membandingkannya dengan aturan lalu lintas masuk Anda. Jika grup keamanan memiliki aturan masuk yang membuka port ke alamat IP publik IPv4 0.0.0.0/0 atau IPv6: :/0, dan port tersebut tidak ditentukan sebagai pengecualian untuk akun Anda, Amazon EMR tidak mengizinkan pengguna membuat klaster.

Jika pengguna memodifikasi aturan grup keamanan untuk klaster yang berjalan di subnet publik agar memiliki aturan akses publik yang melanggar konfigurasi BPA untuk akun Anda, Amazon EMR mencabut aturan baru jika memiliki izin untuk melakukannya. Jika Amazon EMR tidak memiliki izin untuk mencabut aturan, itu akan membuat peristiwa di AWS Health dasbor yang menjelaskan pelanggaran. Untuk memberikan izin aturan pencabutan ke Amazon EMR, lihat. Konfigurasikan Amazon EMR untuk mencabut aturan grup keamanan

Blokir akses publik diaktifkan secara default untuk semua cluster di setiap Wilayah AWS untuk AndaAkun AWS. BPA berlaku untuk seluruh siklus hidup klaster, tetapi tidak berlaku untuk cluster yang Anda buat di subnet pribadi. Anda dapat mengonfigurasi pengecualian ke aturan BPA; port 22 adalah pengecualian secara default. Untuk informasi selengkapnya tentang pengaturan pengecualian, lihatKonfigurasi akses publik blok.

Konfigurasi akses publik blok

Anda dapat memperbarui grup keamanan dan konfigurasi akses publik blok di akun Anda kapan saja.

Anda dapat mengaktifkan dan menonaktifkan pengaturan blokir akses publik (BPA) denganAWS Management Console, AWS Command Line Interface (AWS CLI), dan API EMR Amazon. Pengaturan berlaku di seluruh akun Anda berdasarkan Wilayah demi Wilayah. Untuk menjaga keamanan klaster, kami sarankan Anda menggunakan BPA.

New console

> **Note**
>
> Kami telah mendesain ulang konsol EMR Amazon agar lebih mudah digunakan. Lihat Apa yang baru dengan konsol? untuk mempelajari tentang perbedaan antara pengalaman konsol lama dan baru.

Untuk mengonfigurasi blokir akses publik dengan konsol baru

1. Masuk keAWS Management Console, lalu buka konsol EMR Amazon di https://console.aws.amazon.com/emr.
2. Di bilah navigasi atas, pilih Wilayah yang ingin Anda konfigurasikan jika belum dipilih.
3. Di bawah EMR pada EC2 di panel navigasi kiri, pilih Blokir akses publik.
4. Di bawah Pengaturan akses publik blok, selesaikan langkah-langkah berikut.

Untuk…	Melakukan ini...
Aktifkan atau Nonaktifkan akses publik blok	Pilih Edit, pilih Aktifkan atau Matikan sesuai kebutuhan, lalu pilih Simpan.

Untuk…	Melakukan ini...
Edit port di daftar pengecualian	1. Pilih Edit dan temukan bagian Pengecualian rentang Port. 2. Untuk menambahkan port ke daftar pengecualian, memilih Menambahkan rentang port dan masukkan port baru atau rentang port. Ulangi untuk setiap port atau rentang port untuk menambahkan. 3. Untuk menghapus port atau rentang port, pilih Hapus di samping entri dalam daftar rentang port. 4. Pilih Simpan.

Old console

Untuk melihat konfigurasi blokir akses publik dengan konsol lama

1. Buka konsol EMR Amazon di https://console.aws.amazon.com/emr.
2. Di bilah navigasi atas, verifikasi bahwa Wilayah yang ingin Anda konfigurasikan dipilih.
3. Memilih Akses publik blok.
4. Di bawah Pengaturan akses publik blok, selesaikan langkah-langkah berikut.

Untuk…	Melakukan ini...
Aktifkan atau Nonaktifkan akses publik blok	Memilih Ubah, memilih On atau Off sesuai keinginan, lalu pilih tanda untuk memeriksa dan mengonfirmasi.

Untuk…	Melakukan ini...
Edit port di daftar pengecualian	1. Di bawah Pengecualian, memilih Edit. 2. Untuk menambahkan port ke daftar pengecualian, memilih Menambahkan rentang port dan masukkan port baru atau rentang port. Ulangi untuk setiap port atau rentang port untuk menambahkan. 3. Untuk menghapus port atau rentang port, memilih x di sebelah entri di daftar Rentang port. 4. Memilih Simpan Perubahan.

AWS CLI

Untuk mengonfigurasi akses publik blok menggunakan AWS CLI

- Gunakan perintah `aws emr put-block-public-access-configuration` untuk mengonfigurasi akses publik blok seperti yang ditunjukkan di contoh berikut.

Untuk…	Melakukan ini...
Aktifkan akses publik blok di	Atur `BlockPublicSecurityGroupRules` untuk `true` seperti yang ditunjukkan di contoh berikut. Untuk meluncurkan klaster, tidak ada grup keamanan yang terkait dengan klaster dapat memiliki aturan inbound yang mengizinkan akses publik. `aws emr put-block-public-access-configuration --block-public-access-configuration BlockPublicSecurityGroupRules=true`
Nonaktifkan akses publik blok	Atur `BlockPublicSecurityGroupRules` untuk `false` seperti yang ditunjukkan di contoh berikut. Grup keamanan yang terkait dengan sebuah klaster dapat memiliki aturan inbound yang mengizinkan akses publik pada setiap port. Kami tidak merekomendasikan konfigurasi ini. `aws emr put-block-public-access-configuration --block-public-access-configuration BlockPublicSecurityGroupRules=false`

Untuk…	Melakukan ini...
Aktifkan akses publik blok dan tentukan port sebagai pengecualian	Contoh berikut mengaktifkan akses publik blok, dan menentukan Port 22 dan Port 100-101 sebagai pengecualian. Hal ini mengizinkan klaster yang akan dibuat jika grup keamanan terkait memiliki aturan inbound yang mengizinkan akses publik pada Port 22, Port 100, atau Port 101. `aws emr put-block-public-access-configuration --block-public-access-configuration '{ "BlockPublicSecurityGroupRules": true, "PermittedPublicSecurityGroupRuleRanges": [ { "MinRange": 22, "MaxRange": 22 }, { "MinRange": 100, "MaxRange": 101 } ] }'`

Konfigurasikan Amazon EMR untuk mencabut aturan grup keamanan

Amazon EMR memerlukan izin untuk mencabut aturan grup keamanan dan mematuhi konfigurasi blokir akses publik Anda. Anda dapat menggunakan salah satu pendekatan berikut untuk memberikan izin kepada Amazon EMR yang dibutuhkannya:

- (Disarankan) Lampirkan kebijakan `AmazonEMRServicePolicy_v2` terkelola ke peran layanan. Untuk informasi selengkapnya, lihat Peran layanan untuk Amazon EMR (peran EMR).
- Buat kebijakan inline baru yang memungkinkan `ec2:RevokeSecurityGroupIngress` tindakan pada grup keamanan. Untuk informasi selengkapnya tentang cara mengubah kebijakan izin peran, lihat Memodifikasi kebijakan izin peran dengan Konsol IAM, AWSAPI, dan AWS CLIdalam Panduan Pengguna IAM.

Selesaikan blokir pelanggaran akses publik

Jika terjadi pelanggaran akses publik pemblokiran, Anda dapat menguranginya dengan salah satu tindakan berikut:

- Jika Anda ingin mengakses antarmuka web di cluster Anda, gunakan salah satu opsi yang dijelaskan Melihat antarmuka web yang di-host pada klaster Amazon EMR untuk mengakses antarmuka melalui SSH (port 22).
- Untuk memungkinkan lalu lintas ke cluster dari alamat IP tertentu daripada dari alamat IP publik, tambahkan aturan grup keamanan. Untuk informasi selengkapnya, lihat Menambahkan aturan ke grup keamanan di Panduan Memulai Amazon EC2.
- (Tidak disarankan) Anda dapat mengonfigurasi pengecualian Amazon EMR BPA untuk menyertakan port atau rentang port yang diinginkan. Saat Anda menentukan pengecualian BPA, Anda memperkenalkan risiko dengan port yang tidak dilindungi. Jika Anda berencana untuk menentukan pengecualian, Anda harus menghapus pengecualian segera setelah tidak lagi diperlukan. Untuk informasi selengkapnya, lihat Konfigurasi akses publik blok.

Identifikasi cluster yang terkait dengan aturan grup keamanan

Anda mungkin perlu mengidentifikasi semua cluster yang terkait dengan aturan grup keamanan tertentu, atau untuk menemukan aturan grup keamanan untuk klaster tertentu.

- Jika Anda mengetahui grup keamanan, maka Anda dapat mengidentifikasi cluster terkait jika Anda menemukan antarmuka jaringan untuk grup keamanan. Untuk informasi selengkapnya, lihat Bagaimana cara menemukan sumber daya yang terkait dengan grup keamanan Amazon EC2? padaAWS re:Post. Instans Amazon EC2 yang dilampirkan ke antarmuka jaringan ini akan ditandai dengan ID cluster tempat mereka berada.
- Jika Anda ingin menemukan grup keamanan untuk klaster yang dikenal, ikuti langkah-langkahnyaMelihat status dan detail klaster. Anda dapat menemukan grup keamanan untuk cluster di Jaringan dan panel keamanan di konsol, atau di `Ec2InstanceAttributes` bidang dariAWS CLI.

Validasi kepatuhan untuk Amazon EMR

Auditor pihak ke tiga menilai keamanan dan kepatuhan Amazon EMR sebagai bagian dari beberapa program kepatuhan AWS. Mencakup SOC, PCI, FedRAMP, HIPAA, dan lainnya.

Untuk daftar layanan AWS dalam cakupan program kepatuhan tertentu, lihat Layanan AWS dalam Cakupan berdasarkan Program Kepatuhan. Untuk informasi umum, lihat Program Kepatuhan AWS.

Anda bisa mengunduh laporan audit pihak ke tiga menggunakan AWS Artifact. Untuk informasi selengkapnya, lihat Mengunduh laporan di AWS Artifact.

Tanggung jawab kepatuhan Anda ketika menggunakan Amazon EMR ditentukan oleh sensitivitas data Anda, tujuan kepatuhan korporasi Anda, serta peraturan perundangan yang berlaku. Jika penggunaan Amazon EMR Anda tunduk pada kepatuhan standar seperti HIPAA, PCI, atau FedRAMP, AWS menyediakan sumber daya untuk membantu:

- Panduan Quick Start Keamanan dan Kepatuhan – Panduan deployment ini membahas pertimbangan arsitektur dan memberikan langkah untuk men-deploy lingkungan dasar yang berfokus pada keamanan dan kepatuhan di AWS..
- Perancangan untuk laporan resmi Keamanan dan Kepatuhan HIPAA – Laporan resmi ini menjelaskan cara korporasi dapat menggunakan AWSuntuk membuat aplikasi yang sesuai dengan HIPAA.
- Sumber daya kepatuhan AWS – Kumpulan buku kerja dan panduan ini mungkin berlaku untuk industri dan lokasi Anda.
- AWS Config – Layanan AWS ini menilai seberapa baik konfigurasi sumber daya Anda mematuhi praktik internal, pedoman industri, dan peraturan.
- AWS Security Hub – Layanan AWS ini memberikan tampilan komprehensif atas syarat keamanan Anda di AWS yang membantu Anda memeriksa kepatuhan Anda dengan standar dan praktik terbaik industri keamanan.

Ketahanan di Amazon EMR

Infrastruktur global AWS dibangun di sekitar AWS Wilayah dan Availability Zone.AWS Wilayah menyediakan beberapa Availability Zone yang terpisah dan terisolasi secara fisik, yang terhubung dengan jaringan yang memiliki latensi rendah, throughput tinggi, dan sangat berkelebihan. Dengan Availability Zone, Anda dapat merancang dan mengoperasikan aplikasi dan basis data yang secara otomatis melakukan fail over di antara Availability Zone tanpa gangguan. Availability Zone memiliki ketersediaan yang lebih baik, menoleransi kegagalan, dan dapat diskalakan dibandingkan satu atau beberapa infrastruktur pusat data tradisional.

Untuk informasi selengkapnya tentang AWS Wilayah dan Availability Zone, lihat AWS infrastruktur global.

Selain infrastruktur global AWS, Amazon EMR memberi penawaran beberapa fitur untuk membantu support ketahanan data dan kebutuhan backup Anda.

- Integrasi dengan Amazon S3 melalui EMRFS
- Support untuk beberapa node master

Keamanan infrastruktur di Amazon EMR

Sebagai layanan terkelola, Amazon EMR dilindungi oleh prosedur keamanan jaringan global AWS yang dijelaskan dalam laporan resmi Amazon Web Services: Gambaran Umum Proses Keamanan.

Anda dapat menggunakan perintah API yang dipublikasikan AWS untuk mengakses Amazon EMR melalui jaringan. Klien harus support Keamanan Lapisan Pengangkutan (TLS) 1.2. Klien juga harus support suite cipher dengan perfect forward secrecy (PFS) seperti Ephemeral Diffie-Hellman (DHE) atau Elliptic Curve Ephemeral Diffie-Hellman (ECDHE). Sebagian besar sistem modern seperti Java 7 dan sistem yang lebih baru mendukung mode ini.

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

Topik

- Connect ke Amazon EMR menggunakan VPC endpoint antar muka

Connect ke Amazon EMR menggunakan VPC endpoint antar muka

Anda dapat terhubung langsung ke Amazon EMR menggunakan antarmuka VPC endpoint (AWS PrivateLink) di Virtual Private Cloud (VPC) Anda alih-alih terhubung melalui internet. Ketika Anda menggunakan antarmuka VPC endpoint, komunikasi antara VPC dan Amazon EMR dilakukan sepenuhnya di jaringan AWS. Setiap VPC endpoint diwakili oleh satu Antarmuka jaringan elastis(ENIS) atau lebih dengan alamat IP privat di subnet VPC Anda.

VPC endpoint antarmuka menghubungkan VPC Anda langsung ke Amazon EMR tanpa gateway internet, perangkat NAT, koneksi VPN, atau koneksi AWS Direct Connect. Instans di VPC Anda tidak memerlukan alamat IP publik untuk berkomunikasi dengan API Amazon EMR.

Untuk menggunakan Amazon EMR melalui VPC Anda, Anda harus connect dari sebuah instans yang ada di VPC atau menghubungkan jaringan privat Anda ke VPC Anda dengan menggunakan Jaringan Pribadi Virtual (VPN) Amazon atau AWS Direct Connect. Untuk informasi tentang Amazon VPN, lihat Koneksi VPN di Panduan Pengguna Amazon Virtual Private Cloud. Untuk informasi tentang AWS Direct Connect, lihat Membuat koneksi di AWS Direct Connect Panduan Pengguna.

Anda dapat membuat VPC endpoint antarmuka untuk connect ke Amazon EMR menggunakan AWS konsol atau perintah AWS Command Line Interface (AWS CLI). Untuk informasi selengkapnya, lihat Membuat titik akhir antarmuka.

Setelah Anda membuat VPC endpoint antarmuka, jika Anda mengaktifkan nama host DNS privat untuk endpoint, endpoint Amazon EMR default menyelesaikan VPC endpoint Anda. Endpoint nama layanan default untuk Amazon EMR adalah dalam format berikut.

```
elasticmapreduce.Region.amazonaws.com
```

Jika Anda tidak mengaktifkan nama host DNS privat, Amazon VPC menyediakan endpoint DNS yang dapat Anda gunakan dalam format berikut.

```
VPC_Endpoint_ID.elasticmapreduce.Region.vpce.amazonaws.com
```

Untuk informasi selengkapnya, lihat Antarmuka VPC endpoint (AWS PrivateLink) dalam Panduan Pengguna Amazon VPC.

Amazon EMR mendukung panggilan ke semua tindakan API di dalam VPC Anda.

Anda dapat melampirkan kebijakan VPC endpoint ke VPC endpoint untuk mengontrol akses untuk utama IAM. Anda juga dapat mengasosiasi grup keamanan dengan VPC endpoint untuk mengontrol akses masuk dan keluar berdasarkan asal dan tujuan lalu lintas jaringan, seperti rentang alamat IP. Untuk informasi selengkapnya, lihat Mengendalikan akses ke layanan dengan VPC endpoint.

Buat Kebijakan VPC endpoint untuk Amazon EMR

Anda dapat membuat kebijakan untuk Amazon VPC endpoint untuk Amazon EMR untuk menentukan hal berikut:

- Utama-utama yang dapat melakukan tindakan
- Tindakan yang dapat dilakukan
- Sumber daya yang dapat digunakan untuk mengambil tindakan

Untuk informasi selengkapnya, lihat Mengendalikan akses ke layanan dengan VPC endpoint di Panduan Pengguna Amazon VPC.

Example – Kebijakan VPC endpoint untuk tolak semua akses dari akun AWS tertentu

Kebijakan VPC endpoint berikut tolak akun semua akses AWS *123456789012* ke sumber daya menggunakan titik akhir.

```
{
    "Statement": [
        {
            "Action": "*",
            "Effect": "Allow",
            "Resource": "*",
            "Principal": "*"
        },
        {
            "Action": "*",
            "Effect": "Deny",
            "Resource": "*",
            "Principal": {
                "AWS": [
                    "123456789012"
                ]
            }
        }
    ]
}
```

Example – Kebijakan VPC endpoint untuk mengizinkan akses VPC hanya ke utama IAM tertentu (pengguna)

Kebijakan titik akhir VPC berikut memungkinkan akses penuh hanya ke pengguna lijuan di akun 123456789012. AWS Semua utama IAM lainnya ditolak akses menggunakan titik akhir.

```
{
    "Statement": [
        {
            "Action": "*",
            "Effect": "Allow",
            "Resource": "*",
            "Principal": {
                "AWS": [
                    "arn:aws:iam::123456789012:user/lijuan"
```

```
                ]
            }
        }]
}
```

Example – Kebijakan VPC endpoint untuk mengizinkan operasi EMR hanya-baca

Kebijakan VPC endpoint berikut mengizinkan hanya AWS akun *123456789012* untuk melakukan tindakan Amazon EMR yang ditentukan.

Tindakan yang ditentukan memberikan setara dengan akses hanya-baca untuk Amazon EMR. Semua tindakan lain pada VPC ditolak untuk akun yang ditentukan. Semua akun lain ditolak akses apa pun. Untuk daftar tindakan Amazon EMR, lihat Tindakan, sumber daya, dan kunci syarat untuk Amazon EMR.

```
{
    "Statement": [
        {
            "Action": [
                "elasticmapreduce:DescribeSecurityConfiguration",
                "elasticmapreduce:GetBlockPublicAccessConfiguration",
                "elasticmapreduce:ListBootstrapActions",
                "elasticmapreduce:ViewEventsFromAllClustersInConsole",
                "elasticmapreduce:ListSteps",
                "elasticmapreduce:ListInstanceFleets",
                "elasticmapreduce:DescribeCluster",
                "elasticmapreduce:ListInstanceGroups",
                "elasticmapreduce:DescribeStep",
                "elasticmapreduce:ListInstances",
                "elasticmapreduce:ListSecurityConfigurations",
                "elasticmapreduce:DescribeEditor",
                "elasticmapreduce:ListClusters",
                "elasticmapreduce:ListEditors"
            ],
            "Effect": "Allow",
            "Resource": "*",
            "Principal": {
                "AWS": [
                    "123456789012"
                ]
            }
        }
    ]
```

```
}
```

Example – Kebijakan VPC endpoint menolak akses ke klaster tertentu

Kebijakan VPC endpoint berikut mengizinkan akses penuh untuk semua akun dan utama, namun menolak akses AWS akun*123456789012* apapun untuk tindakan yang dilakukan pada klaster Amazon EMR dengan ID klaster *j-A1B2CD34EF5G*. Tindakan Amazon EMR lain yang tidak support izin tingkat sumber daya untuk klaster masih diizinkan. Untuk daftar tindakan Amazon EMR dan tipe sumber daya yang sesuai, lihat Tindakan, sumber daya, dan kunci syarat untuk Amazon EMR.

```
{
    "Statement": [
        {
            "Action": "*",
            "Effect": "Allow",
            "Resource": "*",
            "Principal": "*"
        },
        {
            "Action": "*",
            "Effect": "Deny",
            "Resource": "arn:aws:elasticmapreduce:us-west-2:123456789012:cluster/j-
A1B2CD34EF5G",
            "Principal": {
                "AWS": [
                    "123456789012"
                ]
            }
        }
    ]
}
```

Mengelola klaster

Setelah Anda meluncurkan klaster, Anda dapat memantau dan mengelolanya. Amazon EMR menyediakan beberapa alat yang dapat Anda gunakan untuk terhubung ke dan mengontrol klaster Anda.

Topik

Connect ke sebuah cluster

Ketika Anda menjalankan klaster Amazon EMR, seringkali yang perlu Anda lakukan hanya menjalankan aplikasi untuk menganalisis data Anda dan kemudian mengumpulkan output dari bucket Amazon S3. Di lain waktu, Anda mungkin ingin berinteraksi dengan node utama saat cluster sedang berjalan. Misalnya, Anda mungkin ingin terhubung ke node utama untuk menjalankan kueri interaktif, memeriksa file log, men-debug masalah dengan cluster, memantau kinerja menggunakan aplikasi seperti Ganglia yang berjalan pada node utama, dan sebagainya. Bagian berikut menjelaskan teknik yang dapat Anda gunakan untuk terhubung ke simpul utama.

Dalam kluster EMR, node utama adalah instans Amazon EC2 yang mengoordinasikan instans EC2 yang berjalan sebagai node tugas dan inti. Node utama mengekspos nama DNS publik yang dapat Anda gunakan untuk menghubungkannya. Secara default, Amazon EMR membuat aturan grup keamanan untuk node utama, dan untuk node inti dan tugas, yang menentukan cara Anda mengakses node.

> **Note**
>
> Anda dapat terhubung ke node utama hanya saat cluster sedang berjalan. Ketika cluster berakhir, instans EC2 bertindak sebagai node utama dihentikan dan tidak lagi tersedia.

Untuk terhubung ke node utama, Anda juga harus mengautentikasi ke cluster. Anda dapat menggunakan Kerberos untuk autentikasi, atau menentukan kunci privat pasangan kunci Amazon EC2 ketika Anda meluncurkan klaster. Untuk informasi selengkapnya tentang mengkonfigurasi Kerberos, dan kemudian menyambungkan, lihat Gunakan Kerberos untuk otentikasi dengan Amazon EMR. Ketika Anda memulai sebuah klaster dari konsol, kunci privat pasangan kunci Amazon EC2 ditentukan dalam bagian Keamanan dan Akses pada halaman Buat Klaster.

Secara default, grup keamanan ElasticMapReduce -master tidak mengizinkan akses SSH masuk. Anda mungkin perlu menambahkan aturan masuk yang mengizinkan akses SSH (TCP port 22) dari sumber tempat Anda ingin memiliki akses. Untuk informasi selengkapnya tentang mengubah aturan grup keamanan, lihat Menambahkan aturan ke grup keamanan dalam Panduan Pengguna Amazon EC2 untuk Instans Linux.

Important

Jangan mengubah aturan yang tersisa di grup keamanan ElasticMapReduce -master. Memodifikasi aturan ini dapat mengganggu operasi klaster.

Topik

- Sebelum menyambungkan: Otorisasi lalu lintas masuk
- Connect ke node utama menggunakan SSH

Sebelum menyambungkan: Otorisasi lalu lintas masuk

Sebelum Anda menyambungkan ke klaster Amazon EMR, Anda harus mengotorisasi SSH lalu lintas masuk (port 22) dari klien tepercaya seperti alamat IP komputer Anda. Untuk melakukannya, edit aturan grup keamanan terkelola untuk simpul yang ingin Anda sambungkan. Misalnya, petunjuk berikut menunjukkan cara menambahkan aturan masuk untuk akses SSH ke grup keamanan ElasticMapReduce -master default.

Untuk informasi selengkapnya tentang menggunakan grup keamanan dengan Amazon EMR, lihat Mengendalikan lalu lintas jaringan dengan grup keamanan.

New console

Untuk memberikan akses SSH sumber tepercaya ke grup keamanan utama dengan konsol baru

Untuk mengedit grup keamanan, Anda harus memiliki izin untuk mengelola grup keamanan untuk VPC tempat klaster berada. Untuk informasi selengkapnya, lihat Mengubah Izin untuk pengguna dan Kebijakan Contoh yang memungkinkan mengelola grup keamanan EC2 di Panduan Pengguna IAM.

1. Masuk keAWS Management Console, dan buka konsol EMR Amazon di https://console.aws.amazon.com/emr.
2. Di bawah EMR pada EC2 di panel navigasi kiri, pilih Cluster, lalu pilih cluster yang ingin Anda perbarui. Ini membuka halaman detail cluster. Tab Properti di halaman ini akan dipilih sebelumnya.
3. Di bawah Jaringan di tab Properties, pilih panah di sebelah grup keamanan EC2 (firewall) untuk memperluas bagian ini. Di bawah Simpul utama, pilih tautan grup keamanan. Ini membuka konsol EC2.
4. Pilih tab Aturan masuk dan kemudian pilih Edit aturan masuk.
5. Memeriksa aturan masuk yang mengizinkan akses publik dengan pengaturan berikut. Jika ada, pilih Hapus untuk menghapusnya.

   - Tipe

     SSH
   - Pelabuhan

     22
   - Sumber

     Kustom 0.0.0.0/0

> ⚠ Warning
>
> Sebelum Desember 2020, grup keamanan ElasticMapReduce -master memiliki aturan yang telah dikonfigurasi sebelumnya untuk mengizinkan lalu lintas masuk di Port 22 dari semua sumber. Aturan ini dibuat untuk menyederhanakan koneksi

SSH awal ke node utama. Kami sangat menyarankan agar Anda menghapus aturan masuk ini dan membatasi lalu lintas ke sumber tepercaya.

6. Gulir ke bagian bawah daftar aturan dan pilih Tambahkan Aturan.
7. Untuk Jenis, pilih SSH. Pilihan ini secara otomatis memasuki TCP untuk Protokol dan 22 untuk Rentang Port.
8. Untuk sumber, pilih IP Saya untuk secara otomatis menambahkan alamat IP Anda sebagai alamat sumber. Anda juga dapat menambahkan berbagai alamat IP klien tepercaya kustom, atau membuat aturan tambahan untuk klien lain. Banyak lingkungan jaringan mengalokasikan alamat IP secara dinamis, jadi Anda mungkin perlu memperbarui alamat IP Anda untuk klien tepercaya di masa mendatang.
9. Pilih Simpan.
10. Secara opsional kembali ke Langkah 3, pilih Core dan node tugas, dan ulangi Langkah 4 - 8. Ini memberikan akses klien SSH node inti dan tugas.

Old console

Untuk memberikan akses SSH sumber tepercaya ke grup keamanan utama dengan konsol lama

Untuk mengedit grup keamanan, Anda harus memiliki izin untuk mengelola grup keamanan untuk VPC tempat klaster berada. Untuk informasi selengkapnya, lihat Mengubah Izin untuk pengguna dan Kebijakan Contoh yang memungkinkan mengelola grup keamanan EC2 di Panduan Pengguna IAM.

1. Arahkan ke konsol EMR Amazon baru dan pilih Beralih ke konsol lama dari navigasi samping. Untuk informasi selengkapnya tentang apa yang diharapkan saat beralih ke konsol lama, lihat Menggunakan konsol lama.
2. Pilih Klaster. Pilih Nama cluster yang ingin Anda modifikasi.
3. Pilih tautan Grup keamanan untuk Master di bawah Keamanan dan akses.
4. Pilih ElasticMapReduce-master dari daftar.
5. Pilih tab Aturan masuk dan kemudian Edit aturan masuk.
6. Memeriksa aturan masuk yang mengizinkan akses publik dengan pengaturan berikut. Jika ada, pilih Hapus untuk menghapusnya.

   - Tipe

SSH

- Pelabuhan

  22

- Sumber

  Kustom 0.0.0.0/0

> **Warning**
>
> Sebelum Desember 2020, grup keamanan ElasticMapReduce -master memiliki aturan yang telah dikonfigurasi sebelumnya untuk mengizinkan lalu lintas masuk di Port 22 dari semua sumber. Aturan ini dibuat untuk menyederhanakan koneksi SSH awal ke node utama. Kami sangat menyarankan agar Anda menghapus aturan masuk ini dan membatasi lalu lintas ke sumber tepercaya.

7. Gulir ke bagian bawah daftar aturan dan pilih Tambahkan Aturan.
8. Untuk Jenis, pilih SSH.

   Memilih SSH secara otomatis memasuki TCP untuk Protokol dan 22 untuk Rentang Port.
9. Untuk sumber, pilih IP Saya untuk secara otomatis menambahkan alamat IP Anda sebagai alamat sumber. Anda juga dapat menambahkan berbagai alamat IP klien tepercaya kustom, atau membuat aturan tambahan untuk klien lain. Banyak lingkungan jaringan mengalokasikan alamat IP secara dinamis, jadi Anda mungkin perlu memperbarui alamat IP Anda untuk klien tepercaya di masa mendatang.
10. Pilih Simpan.
11. Secara opsional, pilih ElasticMapReduce-slave dari daftar dan ulangi langkah-langkah di atas untuk memungkinkan klien SSH mengakses node inti dan tugas.

Connect ke node utama menggunakan SSH

Secure Shell (SSH) adalah protokol jaringan yang dapat Anda gunakan untuk membuat sambungan yang aman ke komputer jarak jauh. Setelah Anda membuat sambungan, terminal pada komputer lokal Anda berperilaku seolah-olah berjalan pada komputer jarak jauh. Perintah yang Anda keluarkan

secara lokal dijalankan di komputer jarak jauh, dan output perintah dari komputer jarak jauh muncul di jendela terminal Anda.

Saat Anda menggunakan SSH dengan AWS, Anda tersambung ke instans EC2, yang merupakan server virtual yang berjalan di cloud. Saat bekerja dengan Amazon EMR, penggunaan SSH yang paling umum adalah terhubung ke instans EC2 yang bertindak sebagai simpul utama cluster.

Menggunakan SSH untuk terhubung ke node utama memberi Anda kemampuan untuk memantau dan berinteraksi dengan cluster. Anda dapat mengeluarkan perintah Linux pada node utama, menjalankan aplikasi seperti Hive dan Pig secara interaktif, menelusuri direktori, membaca file log, dan sebagainya. Anda juga dapat membuat terowongan di koneksi SSH Anda untuk melihat antarmuka web yang dihosting di node utama. Untuk informasi selengkapnya, lihat Melihat antarmuka web yang di-host pada klaster Amazon EMR.

Untuk terhubung ke node primer menggunakan SSH, Anda memerlukan nama DNS publik dari node utama. Selain itu, grup keamanan yang terkait dengan node utama harus memiliki aturan masuk yang memungkinkan lalu lintas SSH (TCP port 22) dari sumber yang mencakup klien tempat koneksi SSH berasal. Anda mungkin perlu menambahkan aturan untuk mengizinkan koneksi SSH dari klien Anda. Untuk informasi selengkapnya tentang mengubah aturan grup keamanan, lihat Mengendalikan lalu lintas jaringan dengan grup keamanan dan Menambahkan aturan ke grup keamanan dalam Panduan Pengguna Amazon EC2 untuk Instans Linux.

Mengambil nama DNS publik dari node utama

Anda dapat mengambil nama DNS publik utama menggunakan konsol EMR Amazon dan. AWS CLI

> **Note**
>
> Kami telah mendesain ulang konsol EMR Amazon agar lebih mudah digunakan. Lihat Apa yang baru dengan konsol? untuk mempelajari tentang perbedaan antara pengalaman konsol lama dan baru.

New console

Untuk mengambil nama DNS publik dari node utama dengan konsol baru

1. Masuk keAWS Management Console, dan buka konsol EMR Amazon di https://console.aws.amazon.com/emr.

2. Di bawah EMR pada EC2 di panel navigasi kiri, pilih Clusters, lalu pilih cluster tempat Anda ingin mengambil nama DNS publik.
3. Perhatikan nilai DNS publik simpul primer di bagian Ringkasan halaman detail klaster.

Old console

Untuk mengambil nama DNS publik dari node utama dengan konsol lama

1. Arahkan ke konsol EMR Amazon baru dan pilih Beralih ke konsol lama dari navigasi samping. Untuk informasi selengkapnya tentang apa yang diharapkan saat beralih ke konsol lama, lihat Menggunakan konsol lama.
2. Pada halaman Daftar Klaster, pilih tautan untuk klaster Anda.
3. Perhatikan nilai DNS publik utama yang muncul di bagian Ringkasan dari halaman Detail Klaster.

> **Note**
>
> Anda juga dapat memilih link SSH untuk instruksi tentang membuat koneksi SSH dengan node utama.

CLI

Untuk mengambil nama DNS publik dari node utama dengan AWS CLI

1. Untuk mengambil pengidentifikasi klaster, ketik perintah berikut.

```
aws emr list-clusters
```

Output mencantumkan klaster Anda termasuk ID klaster. Perhatikan ID klaster untuk klaster yang Anda hubungkan.

```
"Status": {
    "Timeline": {
        "ReadyDateTime": 1408040782.374,
        "CreationDateTime": 1408040501.213
    },
    "State": "WAITING",
```

```
    "StateChangeReason": {
        "Message": "Waiting after step completed"
    }
},
"NormalizedInstanceHours": 4,
"Id": "j-2AL4XXXXXX5T9",
"Name": "My cluster"
```

2. Untuk mencantumkan instance klaster termasuk nama DNS publik untuk klaster, ketikkan salah satu perintah berikut. Ganti *j-2AL4XXXXXX5T9* dengan ID klaster yang dikembalikan oleh perintah sebelumnya.

```
aws emr list-instances --cluster-id j-2AL4XXXXXX5T9
```

Atau:

```
aws emr describe-cluster --cluster-id j-2AL4XXXXXX5T9
```

Output mencantumkan instans klaster termasuk nama DNS dan alamat IP. Perhatikan nilai untuk `PublicDnsName`.

```
"Status": {
    "Timeline": {
        "ReadyDateTime": 1408040779.263,
        "CreationDateTime": 1408040515.535
    },
    "State": "RUNNING",
    "StateChangeReason": {}
},
"Ec2InstanceId": "i-e89b45e7",
"PublicDnsName": "ec2-###-##-##-###.us-west-2.compute.amazonaws.com"

"PrivateDnsName": "ip-###-##-##-###.us-west-2.compute.internal",
"PublicIpAddress": "##.###.###.##",
"Id": "ci-12XXXXXXXXFMH",
"PrivateIpAddress": "###.##.#.###"
```

Untuk informasi selengkapnya, lihat Perintah Amazon EMR dalam AWS CLI.

Connect ke node utama menggunakan SSH dan kunci privat Amazon EC2 di Linux, Unix, dan Mac OS X

Untuk membuat koneksi SSH yang diautentikasi dengan file kunci privat, Anda perlu menentukan kunci privat pasangan kunci Amazon EC2 ketika Anda meluncurkan suatu klaster. Untuk informasi selengkapnya tentang mengakses pasangan kunci Anda, lihat Pasangan kunci Amazon EC2 dalam Panduan Pengguna Amazon EC2 untuk Instans Linux.

Komputer Linux Anda kemungkinan besar memiliki klien SSH secara default. Sebagai contoh, OpenSSH dipasang pada kebanyakan sistem operasi Linux, Unix, dan MacOS. Anda dapat memeriksa klien SSH dengan mengetik ssh di baris perintah. Jika komputer Anda tidak mengenali perintah, instal klien SSH untuk terhubung ke node utama. Proyek OpenSSH menyediakan implementasi gratis rangkaian lengkap alat SSH. Untuk informasi selengkapnya, lihat situs web OpenSSH.

Instruksi berikut menunjukkan membuka koneksi SSH ke simpul utama Amazon EMR di Linux, Unix, dan Mac OS X.

Untuk mengonfigurasi izin file kunci privat pasangan kunci

Sebelum Anda dapat menggunakan kunci privat pasangan kunci Amazon EC2 Anda untuk membuat koneksi SSH, Anda harus mengatur izin pada file `.pem` agar hanya pemilik kunci yang memiliki izin untuk mengakses file. Hal ini diperlukan untuk membuat koneksi SSH menggunakan terminal atau AWS CLI.

1. Pastikan Anda telah mengizinkan lalu lintas SSH masuk. Untuk melihat instruksi, lihat Sebelum menyambungkan: Otorisasi lalu lintas masuk.
2. Temukan file `.pem` Anda. Instruksi berikut mengasumsikan bahwa file dinamakan `mykeypair.pem` dan disimpan dalam direktori beranda pengguna saat ini.
3. Ketik perintah berikut untuk mengatur izin. Ganti *~/mykeypair.pem* dengan jalur lengkap dan nama file untuk file kunci privat pasangan kunci Anda. Sebagai contoh `C:/Users/<username>/.ssh/mykeypair.pem`.

   ```
   chmod 400 ~/mykeypair.pem
   ```

   Jika Anda tidak mengatur izin pada file `.pem`, Anda akan menerima kesalahan yang menunjukkan bahwa file kunci Anda tidak dilindungi dan kunci akan ditolak. Untuk

menyambungkan, Anda hanya perlu mengatur izin pada file kunci privat pasangan kunci saat pertama kali Anda menggunakannya.

Untuk terhubung ke node utama menggunakan terminal

1. Buka jendela terminal. Pada Mac OS X, pilih Aplikasi > Utilitas > Terminal. Pada distribusi Linux lainnya, terminal biasanya ditemukan di Aplikasi > Aksesori > Terminal.
2. Untuk membuat koneksi ke node utama, ketik perintah berikut. *Ganti ec2 ###-#-#-#-#-##.compute-1.amazonaws.com dengan nama DNS publik utama cluster Anda dan ganti ~/mykeypair.pem dengan path lengkap dan nama file file Anda.* `.pem` Sebagai contoh, `C:/Users/<username>/.ssh/mykeypair.pem`.

   ```
   ssh hadoop@ec2-###-##-##-###.compute-1.amazonaws.com -i ~/mykeypair.pem
   ```

   > **Important**
   >
   > Anda harus menggunakan nama login `hadoop` ketika Anda terhubung ke simpul utama Amazon EMR; jika tidak, Anda mungkin melihat kesalahan yang mirip dengan. `Server refused our key`

3. Muncul peringatan yang menyatakan bahwa keaslian host yang Anda sambungkan tidak dapat diverifikasi. Ketik `yes` untuk melanjutkan.
4. Ketika Anda selesai bekerja pada node utama, ketik perintah berikut untuk menutup koneksi SSH.

   ```
   exit
   ```

Jika Anda mengalami kesulitan dalam menggunakan SSH untuk terhubung ke node utama Anda, lihat Memecahkan masalah menghubungkan ke instans Anda.

Connect ke node utama menggunakan SSH pada Windows

Pengguna Windows dapat menggunakan klien SSH seperti PuTTY untuk terhubung ke node utama. Sebelum menghubungkan ke simpul utama Amazon EMR, Anda harus mengunduh dan menginstal Putty dan PuttyGen. Anda dapat mengunduh keduanya dari halaman unduh PuTTY.

PuTTY tidak secara native mendukung format file kunci privat pasangan kunci (`.pem`) yang dihasilkan oleh Amazon EC2. Anda menggunakan PuTTYgen untuk mengubah file kunci Anda ke format PuTTY yang diperlukan (`.ppk`). Anda harus mengonversi kunci Anda ke format ini (`.ppk`) sebelum mencoba terhubung ke node utama menggunakan PuTTY.

Untuk informasi selengkapnya tentang mengubah kunci Anda, lihat Mengubah kunci privat Anda menggunakan PuTTYgen dalam Panduan Pengguna Amazon EC2 untuk Instans Linux.

Untuk terhubung ke node utama menggunakan PuTTY

1. Pastikan Anda telah mengizinkan lalu lintas SSH masuk. Untuk melihat instruksi, lihat Sebelum menyambungkan: Otorisasi lalu lintas masuk.
2. Buka `putty.exe`. Anda juga dapat meluncurkan PuTTY dari daftar program Windows.
3. Jika perlu, di daftar Kategori, pilih Sesi.
4. Untuk Nama Host (atau alamat IP), ketik `hadoop@` *`MasterPublicDNS`*. Sebagai contoh: `hadoop@`*`ec2-###-##-##-###.compute-1.amazonaws.com`*.
5. Di daftar Kategori, pilih Koneksi > SSH, Autentikasi.
6. Untuk File kunci privat untuk autentikasi, pilih Telusuri dan pilih file `.ppk` yang Anda buat.
7. Pilih Buka lalu Ya untuk mengabaikan pemberitahuan keamanan PuTTY.

   > **Important**
   >
   > Saat masuk ke node utama, ketik `hadoop` jika Anda diminta untuk nama pengguna.

8. Ketika Anda selesai bekerja pada node utama, Anda dapat menutup koneksi SSH dengan menutup PuTTY.

   > **Note**
   >
   > Untuk mencegah koneksi SSH kehabisan waktu, Anda dapat memilih Koneksi dalam daftar Kategori dan pilih opsi Aktifkan TCP_keepalives. Jika Anda memiliki sesi SSH aktif di PuTTY, Anda dapat mengubah pengaturan Anda dengan membuka konteks (klik kanan) untuk bilah judul PuTTY dan memilih Mengubah Pengaturan.

Jika Anda mengalami kesulitan dalam menggunakan SSH untuk terhubung ke node utama Anda, lihat Memecahkan masalah menghubungkan ke instans Anda.

Connect ke node utama menggunakan AWS CLI

Anda dapat membuat koneksi SSH dengan node utama menggunakan AWS CLI pada Windows dan Linux, Unix, dan Mac OS X. Terlepas dari platform, Anda memerlukan nama DNS publik dari node utama dan kunci privat Amazon EC2 key pair Anda. Jika Anda menggunakan AWS CLI di Linux, Unix, atau Mac OS X, Anda juga harus mengatur izin pada file kunci privat (`.pem` atau `.ppk`) seperti yang ditunjukkan dalam Untuk mengonfigurasi izin file kunci privat pasangan kunci.

Untuk terhubung ke node utama menggunakan AWS CLI

1. Pastikan Anda telah mengizinkan lalu lintas SSH masuk. Untuk melihat instruksi, lihat Sebelum menyambungkan: Otorisasi lalu lintas masuk.
2. Untuk mengambil pengidentifikasi klaster, ketik:

   ```
   aws emr list-clusters
   ```

   Output mencantumkan klaster Anda termasuk ID klaster. Perhatikan ID klaster untuk klaster yang Anda hubungkan.

   ```
   "Status": {
       "Timeline": {
           "ReadyDateTime": 1408040782.374,
           "CreationDateTime": 1408040501.213
       },
       "State": "WAITING",
       "StateChangeReason": {
           "Message": "Waiting after step completed"
       }
   },
   "NormalizedInstanceHours": 4,
   "Id": "j-2AL4XXXXXX5T9",
   "Name": "AWS CLI cluster"
   ```

3. Ketik perintah berikut untuk membuka koneksi SSH ke node utama. Pada contoh berikut, ganti *j-2AL4XXXXXX5T9* dengan ID klaster dan ganti *~/mykeypair.key* dengan jalur lengkap dan nama file untuk file `.pem` Anda (untuk Linux, Unix, dan Mac OS X) atau file `.ppk` (untuk Windows). Sebagai contoh `C:\Users\<username>\.ssh\mykeypair.pem`.

   ```
   aws emr ssh --cluster-id j-2AL4XXXXXX5T9 --key-pair-file ~/mykeypair.key
   ```

4. Ketika Anda selesai bekerja pada node utama, tutup AWS CLI jendela.

   Untuk informasi selengkapnya, lihat Perintah Amazon EMR di AWS CLI. Jika Anda mengalami kesulitan dalam menggunakan SSH untuk terhubung ke node utama Anda, lihat Memecahkan masalah menghubungkan ke instans Anda.

Port layanan Amazon EMR

> **Note**
>
> Berikut ini adalah antarmuka dan port layanan untuk komponen di Amazon EMR. Ini bukan daftar lengkap port layanan. Layanan non-default, seperti port SSL dan berbagai jenis protokol, tidak terdaftar.

> **Important**
>
> Berhati-hatilah saat Anda mengedit aturan grup keamanan untuk membuka port. Pastikan untuk menambahkan aturan yang hanya mengizinkan lalu lintas dari klien tepercaya dan terautentikasi untuk protokol dan port yang diperlukan untuk menjalankan beban kerja Anda.

Komponen	Deskripsi layanan	Layanan berjalan secara default	Port	Kunci konfigurasi
Hadoop	HTTP KMS SISA API	Ya	9600	hadoop.kms.http.port
HDFS	Namenode Web UI	Ya	9870	dfs.namenode.http-alamat
	Namenode RPC	Ya	8020	dfs.namenode.rpc-address
	DataNode UI Web	Ya	9864	dfs.datanode.http.address

Komponen	Deskripsi layanan	Layanan berjalan secara default	Port	Kunci konfigurasi
	Datanode HTTP untuk transfer data	Ya	9866	dfs.datanode.address
	Datanode RPC untuk transfer data	Ya	9867	dfs.datanode.ipc.address
Hive	HiveServer2 Penghematan	Ya	10000	hive.server2.thrift.port
	HiveServer2 HTTP	Tidak	10001	hive.server2.thrift.http.port
	HiveServer2 Web UI	Ya	10002	hive.server2.webui.port
	Metastore Sarang	Ya	9083	hive.metastore.port/metastore.thrift.port
	WebHCat	Tidak	50111	templeton.port
	Layanan manajemen daemon LLAP (RPC)	Tidak	15004	hive.llap.management.rpc.port
	Port pengocokan YARN untuk shuffle yang dihosting LLAP-Daemon	Tidak	15551	hive.llap.daemon.yarn.shuffle.port
	RPC daemon LLAP	Tidak	Dinamis	hive.llap.daemon.rpc.port

Komponen	Deskripsi layanan	Layanan berjalan secara default	Port	Kunci konfigurasi
	LLAP daemon Web UI	Tidak	15002	hive.llap .daemon.web.port
	Layanan keluaran daemon LLAP	Tidak	15003	hive.llap .daemon.o utput.service.port
Oozie		Ya	11000	
Tez	Tez UI	Ya	8080	
YARN	Kocok	Ya	13562	mapreduce .shuffle.port
	Pelokalisasi RPC	Ya	8040	yarn.node manager.localizer. address
		Ya	8041	
	Alamat Webapp NM	Ya	8042	yarn.node manager.w ebapp.address
	Aplikasi web RM	Ya	8088	yarn.reso urcemanag er.webapp .address
		Ya	8025	
	Penjadwal	Ya	8030	yarn.reso urcemanag er.scheduler.addre ss

Komponen	Deskripsi layanan	Layanan berjalan secara default	Port	Kunci konfigurasi
	antarmuka manajer aplikasi	Ya	8032	yarn.resourcemanager.address
	Antarmuka admin RM	Ya	8033	yarn.resourcemanager.admin.address
	JobHistory UI Web Server	Ya	19888	mapreduce.jobhistory.webapp.address
	JobHistory Admin Server Web UI	Ya	10033	mapreduce.jobhistory.admin.address
	JobHistory Server (RPC)	Ya	10020	mapreduce.jobhistory.address
	Server Timeline Aplikasi (RPC)	Ya	10200	yarn.timeline-service.address
	Aplikasi Timeline Server HTTP Web UI	Ya	8188	yarn.timeline-service.webapp.address
	Aplikasi Timeline Server HTTPS Web UI	Tidak	8190	yarn.timeline-service.webapp.https.address
		Ya	20888	
Zookeeper	Port klien	Ya	2181	
		Ya	37301	

Komponen	Deskripsi layanan	Layanan berjalan secara default	Port	Kunci konfigurasi
		Ya	8341	

Melihat antarmuka web yang di-host pada klaster Amazon EMR

> **Important**
>
> Anda dapat mengonfigurasi grup keamanan kustom untuk mengizinkan akses masuk ke antarmuka web ini. Perlu diingat bahwa setiap port tempat Anda mengizinkan lalu lintas masuk merupakan potensi kelemahan keamanan. Cermatlah dalam meninjau grup keamanan kustom untuk memastikan bahwa Anda meminimalisir kelemahan. Untuk informasi selengkapnya, lihat Mengendalikan lalu lintas jaringan dengan grup keamanan.

Hadoop dan aplikasi lain yang Anda instal di kluster EMR Anda mempublikasikan antarmuka pengguna sebagai situs web yang di-host di node utama. Untuk alasan keamanan, saat menggunakan Grup Keamanan Terkelola Amazon EMR, situs web ini hanya tersedia di server web lokal node utama. Untuk alasan itu, Anda perlu terhubung ke node utama untuk melihat antarmuka web. Untuk informasi selengkapnya, lihat Connect ke node utama menggunakan SSH. Hadoop juga menerbitkan antarmuka pengguna sebagai situs web yang di-host pada simpul inti dan tugas. Situs web tersebut juga hanya tersedia di server web lokal pada simpul.

Tabel berikut ini mencantumkan antarmuka web yang dapat Anda lihat pada instans klaster. Antarmuka Hadoop ini tersedia pada semua klaster. Untuk antarmuka instance master, ganti *`master-public-dns-name`*dengan DNS publik Master yang tercantum di tab Ringkasan cluster di konsol EMR Amazon. Untuk antarmuka inti dan instance tugas, ganti *`coretask-public-dns-name`*dengan nama DNS Publik yang terdaftar untuk instance. Untuk menemukan nama DNS Publik instans, di konsol EMR Amazon, pilih klaster Anda dari daftar, pilih tab Perangkat Keras, pilih ID grup instans yang berisi instance yang ingin Anda sambungkan, lalu catat nama DNS Publik yang terdaftar untuk instance tersebut.

Nama antarmuka	URI
Server riwayat Flink (EMR versi 5.33 dan yang lebih baru)	http: //:8082/ *master-public-dns-name*
Ganglia	http:///ganglia/*master-public-dns-name*
Hadoop HDFS (versi NameNode EMR pra-6.x)	https: //:50470/ *master-public-dns-name*
Hadoop HDFS NameNode	http: //:50070/ *master-public-dns-name*
Hadoop HDFS DataNode	http: //:50075/ *coretask-public-dns-name*
Hadoop HDFS (NameNode EMR versi 6.x)	https: //:9870/ *master-public-dns-name*
Hadoop HDFS (versi DataNode EMR pra-6.x)	https: //:50475/ *coretask-public-dns-name*
Hadoop HDFS (DataNode EMR versi 6.x)	https: //:9865/ *coretask-public-dns-name*
HBase	http: //:16010/ *master-public-dns-name*
Hue	http: //:8888/ *master-public-dns-name*
JupyterHub	https: //:9443/ *master-public-dns-name*
Livy	http: //:8998/ *master-public-dns-name*
Percikan HistoryServer	http: //:18080/ *master-public-dns-name*
Tez	http: //:8080/tez-ui *master-public-dns-name*
BENANG NodeManager	http: //:8042/ *coretask-public-dns-name*
BENANG ResourceManager	http: //:8088/ *master-public-dns-name*
Zeppelin	http: //:8890/ *master-public-dns-name*

Karena ada beberapa antarmuka khusus aplikasi yang tersedia di simpul utama yang tidak tersedia pada node inti dan tugas, instruksi dalam dokumen ini khusus untuk simpul primer Amazon EMR. Mengakses antarmuka web pada inti dan node tugas dapat dilakukan dengan cara yang sama seperti Anda akan mengakses antarmuka web pada node utama.

Ada beberapa cara Anda dapat mengakses antarmuka web pada node utama. Metode termudah dan tercepat adalah menggunakan SSH untuk terhubung ke node utama dan menggunakan browser berbasis teks, Lynx, untuk melihat situs web di klien SSH Anda. Namun, Lynx adalah peramban berbasis teks dengan antarmuka pengguna terbatas yang tidak dapat menampilkan grafis. Contoh berikut menunjukkan cara membuka ResourceManager antarmuka Hadoop menggunakan Lynx (URL Lynx juga disediakan saat Anda masuk ke node utama menggunakan SSH).

```
lynx http://ip-###-##-##-###.us-west-2.compute.internal:8088/
```

Ada dua opsi yang tersisa untuk mengakses antarmuka web pada node utama yang menyediakan fungsionalitas browser penuh. Pilih salah satu dari berikut:

- Opsi 1 (direkomendasikan untuk pengguna yang lebih teknis): Gunakan klien SSH untuk terhubung ke node utama, konfigurasikan tunneling SSH dengan penerusan port lokal, dan gunakan browser Internet untuk membuka antarmuka web yang dihosting di node utama. Metode ini memungkinkan Anda untuk mengonfigurasi akses antarmuka web tanpa menggunakan proksi SOCKS.
- Opsi 2 (direkomendasikan untuk pengguna baru): Gunakan klien SSH untuk terhubung ke node utama, konfigurasikan tunneling SSH dengan penerusan port dinamis, dan konfigurasikan browser Internet Anda untuk menggunakan add-on seperti untuk Firefox atau Chrome FoxyProxy SwitchyOmega untuk mengelola pengaturan proxy SOCKS Anda. Metode ini memungkinkan Anda secara otomatis memfilter URL berdasarkan pola teks dan membatasi pengaturan proxy ke domain yang cocok dengan bentuk nama DNS node utama. Untuk informasi selengkapnya tentang cara mengonfigurasi FoxyProxy Firefox dan Google Chrome, lihatOpsi 2, bagian 2: Konfigurasikan pengaturan proxy untuk melihat situs web yang dihosting di simpul utama.

> **Note**
>
> Jika Anda memodifikasi port tempat aplikasi berjalan melalui konfigurasi cluster, hyperlink ke port tidak akan diperbarui di konsol EMR Amazon. Ini karena konsol tidak memiliki fungsi untuk membaca `server.port` konfigurasi.

Dengan Amazon EMR versi 5.25.0 atau yang lebih baru, Anda dapat mengakses UI server riwayat Spark dari konsol tanpa mengatur proksi web melalui koneksi SSH. Untuk informasi selengkapnya, lihat Akses satu klik ke server riwayat Spark persisten.

Topik

- Opsi 1: Siapkan terowongan SSH ke simpul utama menggunakan penerusan port lokal
- Opsi 2, bagian 1: Siapkan terowongan SSH ke simpul utama menggunakan penerusan port dinamis
- Opsi 2, bagian 2: Konfigurasikan pengaturan proxy untuk melihat situs web yang dihosting di simpul utama

Opsi 1: Siapkan terowongan SSH ke simpul utama menggunakan penerusan port lokal

Untuk terhubung ke server web lokal pada node utama, Anda membuat terowongan SSH antara komputer Anda dan node utama. Ini juga dikenal sebagai penerusan port. Jika Anda tidak ingin menggunakan proxy SOCKS, Anda dapat mengatur terowongan SSH ke node utama menggunakan penerusan port lokal. Dengan penerusan port lokal, Anda menentukan port lokal yang tidak digunakan yang digunakan untuk meneruskan lalu lintas ke port jarak jauh tertentu di server web lokal node utama.

Menyiapkan terowongan SSH menggunakan penerusan port lokal memerlukan nama DNS publik dari node utama dan file kunci pribadi key pair Anda. Untuk informasi tentang cara menemukan nama DNS publik utama, lihat Untuk mengambil nama DNS publik dari node utama dengan konsol lama. Untuk informasi selengkapnya tentang mengakses pasangan kunci Anda, lihat Pasangan kunci Amazon EC2 dalam Panduan Pengguna Amazon EC2 untuk Instans Linux. Untuk informasi selengkapnya tentang situs yang mungkin ingin Anda lihat di simpul utama, lihatMelihat antarmuka web yang di-host pada klaster Amazon EMR.

Siapkan terowongan SSH ke node utama menggunakan penerusan port lokal dengan OpenSSH

Untuk mengatur sebuah terowongan SSH menggunakan penerusan port lokal di terminal

1. Pastikan Anda telah mengizinkan lalu lintas SSH masuk. Untuk melihat instruksi, lihat Sebelum menyambungkan: Otorisasi lalu lintas masuk.
2. Buka jendela terminal. Pada Mac OS X, pilih Aplikasi > Utilitas > Terminal. Pada distribusi Linux lainnya, terminal biasanya ditemukan di Aplikasi > Aksesori > Terminal.
3. Ketik perintah berikut untuk membuka terowongan SSH pada mesin lokal Anda. Contoh perintah ini mengakses antarmuka ResourceManager web dengan meneruskan lalu lintas pada port

lokal 8157 (port lokal yang tidak digunakan secara acak) ke port 8088 di server web lokal master node.

Dalam perintah, ganti *~/mykeypair.pem* dengan lokasi dan nama `.pem` file Anda dan ganti *ec2-###-##-##-###.compute-1.amazonaws.com* dengan nama DNS publik utama klaster Anda. Untuk mengakses antarmuka web yang berbeda, ganti `8088` dengan nomor port yang sesuai. Misalnya, ganti `8088` dengan

```
ssh -i ~/mykeypair.pem -N -L 8157:ec2-###-##-##-###.compute-1.amazonaws.com:8088 hadoop@ec2-###-##-##-###.compute-1.amazonaws.com
```

`-L` menandakan penggunaan penerusan port lokal yang memungkinkan Anda untuk menentukan port lokal yang digunakan untuk meneruskan data ke port jarak jauh yang teridentifikasi pada server web lokal simpul utama.

Setelah Anda mengeluarkan perintah ini, terminal tetap terbuka dan tidak mengembalikan respons.

4. Untuk membuka antarmuka ResourceManager web di browser Anda, `http://localhost:8157/` ketik bilah alamat.
5. Ketika Anda selesai bekerja dengan antarmuka web pada node utama, tutup jendela terminal.

Opsi 2, bagian 1: Siapkan terowongan SSH ke simpul utama menggunakan penerusan port dinamis

Untuk terhubung ke server web lokal pada node utama, Anda membuat terowongan SSH antara komputer Anda dan node utama. Ini juga dikenal sebagai penerusan port. Jika Anda membuat terowongan SSH menggunakan penerusan port dinamis, semua lalu lintas yang dirutekan ke port lokal yang tidak digunakan tertentu diteruskan ke server web lokal pada node utama. Hal ini menciptakan proksi SOCKS. Anda kemudian dapat mengonfigurasi browser Internet Anda untuk menggunakan add-on seperti FoxyProxy atau SwitchyOmega untuk mengelola pengaturan proxy SOCKS Anda.

Menggunakan add-on manajemen proxy memungkinkan Anda untuk secara otomatis memfilter URL berdasarkan pola teks dan membatasi pengaturan proxy ke domain yang cocok dengan bentuk nama DNS publik node utama. Add-on browser secara otomatis menangani menghidupkan dan mematikan proxy saat Anda beralih antara melihat situs web yang dihosting di node utama, dan yang ada di Internet.

Sebelum memulai, Anda memerlukan nama DNS publik dari node utama dan file kunci pribadi key pair Anda. Untuk informasi tentang cara menemukan nama DNS publik utama, lihatUntuk mengambil nama DNS publik dari node utama dengan konsol lama. Untuk informasi selengkapnya tentang mengakses pasangan kunci Anda, lihat Pasangan kunci Amazon EC2 dalam Panduan Pengguna Amazon EC2 untuk Instans Linux. Untuk informasi selengkapnya tentang situs yang mungkin ingin Anda lihat di simpul utama, lihatMelihat antarmuka web yang di-host pada klaster Amazon EMR.

Siapkan terowongan SSH ke node utama menggunakan penerusan port dinamis dengan OpenSSH

Untuk mengatur terowongan SSH menggunakan penerusan port dinamis dengan OpenSSH

1. Pastikan Anda telah mengizinkan lalu lintas SSH masuk. Untuk melihat instruksi, lihat Sebelum menyambungkan: Otorisasi lalu lintas masuk.
2. Buka jendela terminal. Pada Mac OS X, pilih Aplikasi > Utilitas > Terminal. Pada distribusi Linux lainnya, terminal biasanya ditemukan di Aplikasi > Aksesori > Terminal.
3. Ketik perintah berikut untuk membuka terowongan SSH pada mesin lokal Anda. *Ganti ~/mykeypair.pem dengan lokasi dan nama file .pem file Anda, ganti 8157 dengan nomor port lokal yang tidak digunakan, dan ganti c2 ####-#-#-#-#-#.compute-1.amazonaws.com dengan nama DNS publik utama cluster Anda.*

   ```
   ssh -i ~/mykeypair.pem -N -D 8157 hadoop@ec2-###-##-##-###.compute-1.amazonaws.com
   ```

   Setelah Anda mengeluarkan perintah ini, terminal tetap terbuka dan tidak mengembalikan respons.

   > **Note**
   >
   > `-D`menandakan penggunaan penerusan port dinamis yang memungkinkan Anda menentukan port lokal yang digunakan untuk meneruskan data ke semua port jarak jauh di server web lokal node utama. Penerusan port dinamis membuat proksi SOCKS lokal yang mendengarkan port yang ditentukan dalam perintah.

4. Setelah terowongan aktif, konfigurasikan proksi SOCKS untuk peramban Anda. Untuk informasi selengkapnya, lihat Opsi 2, bagian 2: Konfigurasikan pengaturan proxy untuk melihat situs web yang dihosting di simpul utama.
5. Ketika Anda selesai bekerja dengan antarmuka web pada node utama, tutup jendela terminal.

Mengatur sebuah terowongan SSH menggunakan port forwarding dinamis dengan AWS CLI

Anda dapat membuat koneksi SSH dengan node utama menggunakan AWS CLI pada Windows dan di Linux, Unix, dan Mac OS X. Jika Anda menggunakan AWS CLI di Linux, Unix, atau Mac OS X, Anda harus mengatur izin pada file seperti yang ditunjukkan pada `.pem`. Untuk mengonfigurasi izin file kunci privat pasangan kunci Jika Anda menggunakan AWS CLI pada Windows, PuTTY harus muncul dalam variabel lingkungan jalur atau Anda akan menerima kesalahan seperti OpenSSH atau PuTTY tidak tersedia.

Untuk mengatur terowongan SSH menggunakan penerusan port dinamis dengan AWS CLI

1. Pastikan Anda telah mengizinkan lalu lintas SSH masuk. Untuk melihat instruksi, lihat Sebelum menyambungkan: Otorisasi lalu lintas masuk.
2. Buat koneksi SSH dengan node utama seperti yang ditunjukkan padaConnect ke node utama menggunakan AWS CLI.
3. Untuk mengambil pengidentifikasi klaster, ketik:

   ```
   aws emr list-clusters
   ```

   Output mencantumkan klaster Anda termasuk ID klaster. Perhatikan ID klaster untuk klaster yang Anda hubungkan.

   ```
   "Status": {
       "Timeline": {
           "ReadyDateTime": 1408040782.374,
           "CreationDateTime": 1408040501.213
       },
       "State": "WAITING",
       "StateChangeReason": {
           "Message": "Waiting after step completed"
       }
   },
   "NormalizedInstanceHours": 4,
   "Id": "j-2AL4XXXXXX5T9",
   "Name": "AWS CLI cluster"
   ```

4. Ketik perintah berikut untuk membuka terowongan SSH ke node utama menggunakan penerusan port dinamis. Dalam contoh berikut, ganti *`j-2AL4XXXXXX5T9`* dengan ID klaster dan ganti *`~/mykeypair.key`* dengan lokasi dan nama file untuk file `.pem` Anda (untuk Linux, Unix, dan Mac OS X) atau file `.ppk` (untuk Windows).

```
aws emr socks --cluster-id j-2AL4XXXXXX5T9 --key-pair-file ~/mykeypair.key
```

> **Note**
>
> Perintah socks secara otomatis mengonfigurasi penerusan port dinamis pada port lokal 8157. Saat ini, pengaturan ini tidak dapat diubah.

5. Setelah terowongan aktif, konfigurasikan proksi SOCKS untuk peramban Anda. Untuk informasi selengkapnya, lihat Opsi 2, bagian 2: Konfigurasikan pengaturan proxy untuk melihat situs web yang dihosting di simpul utama.
6. Ketika Anda selesai bekerja dengan antarmuka web pada node utama, tutup AWS CLI jendela.

   Untuk informasi selengkapnya tentang menggunakan perintah Amazon EMR di AWS CLI, lihat https://docs.aws.amazon.com/cli/latest/reference/emr.

Siapkan terowongan SSH ke simpul utama menggunakan PuTTY

Pengguna Windows dapat menggunakan klien SSH seperti PuTTY untuk membuat terowongan SSH ke node utama. Sebelum menghubungkan ke simpul utama Amazon EMR, Anda harus mengunduh dan menginstal Putty dan PuttyGen. Anda dapat mengunduh keduanya dari halaman unduh PuTTY.

PuTTY tidak secara native mendukung format file kunci privat pasangan kunci (`.pem`) yang dihasilkan oleh Amazon EC2. Anda menggunakan PuTTYgen untuk mengubah file kunci Anda ke format PuTTY yang diperlukan (`.ppk`). Anda harus mengonversi kunci Anda ke format ini (`.ppk`) sebelum mencoba terhubung ke node utama menggunakan PuTTY.

Untuk informasi selengkapnya tentang mengubah kunci Anda, lihat Mengubah kunci privat Anda menggunakan PuTTYgen dalam Panduan Pengguna Amazon EC2 untuk Instans Linux.

Untuk mengatur terowongan SSH menggunakan penerusan port dinamis menggunakan PuTTY

1. Pastikan Anda telah mengizinkan lalu lintas SSH masuk. Untuk melihat instruksi, lihat Sebelum menyambungkan: Otorisasi lalu lintas masuk.
2. Klik dua kali `putty.exe` untuk memulai PuTTY. Anda juga dapat meluncurkan PuTTY dari daftar program Windows.

> **Note**
>
> Jika Anda sudah memiliki sesi SSH aktif dengan node utama, Anda dapat menambahkan terowongan dengan mengklik kanan bilah judul PuTTY dan memilih Ubah Pengaturan.

3. Jika perlu, di daftar Kategori, pilih Sesi.
4. Di bidang Nama Host, ketik **hadoop@** *MasterPublicDNS*. Sebagai contoh: **hadoop@***ec2-###-##-##-###.compute-1.amazonaws.com*.
5. Dalam daftar Kategori, perluas Koneksi > SSH, lalu pilih Autentikasi.
6. Untuk File kunci privat untuk autentikasi, pilih Telusuri dan pilih file `.ppk` yang Anda buat.

> **Note**
>
> PuTTY tidak secara native mendukung format file kunci privat pasangan kunci (`.pem`) yang dihasilkan oleh Amazon EC2. Anda menggunakan PuTTYgen untuk mengubah file kunci Anda ke format PuTTY yang diperlukan (`.ppk`). Anda harus mengonversi kunci Anda ke format ini (`.ppk`) sebelum mencoba terhubung ke node utama menggunakan PuTTY.

7. Dalam daftar Kategori, perluas Koneksi > SSH, lalu pilih Terowongan.
8. Dalam bidang Port sumber, ketik `8157` (port lokal tidak terpakai), lalu pilih Menambahkan.
9. Biarkan bidang Tujuan kosong.
10. Pilih opsi Dinamis dan Otomatis.
11. Pilih Buka.
12. Pilih Ya untuk menghilangkan pemberitahuan keamanan PuTTY.

> **Important**
>
> Saat Anda masuk ke node utama, ketik `hadoop` jika Anda diminta untuk nama pengguna.

13. Setelah terowongan aktif, konfigurasikan proksi SOCKS untuk peramban Anda. Untuk informasi selengkapnya, lihat Opsi 2, bagian 2: Konfigurasikan pengaturan proxy untuk melihat situs web yang dihosting di simpul utama.
14. Ketika Anda selesai bekerja dengan antarmuka web pada node utama, tutup jendela Putty.

Opsi 2, bagian 2: Konfigurasikan pengaturan proxy untuk melihat situs web yang dihosting di simpul utama

Jika Anda menggunakan terowongan SSH dengan penerusan port dinamis, Anda harus menggunakan add-on manajemen proksi SOCKS untuk mengendalikan pengaturan proksi di peramban Anda. Menggunakan alat manajemen proxy SOCKS memungkinkan Anda untuk secara otomatis memfilter URL berdasarkan pola teks dan membatasi pengaturan proxy ke domain yang cocok dengan bentuk nama DNS publik node utama. Add-on browser secara otomatis menangani menghidupkan dan mematikan proxy ketika Anda beralih antara melihat situs web yang dihosting di node utama dan yang ada di Internet. Untuk mengelola pengaturan proxy Anda, konfigurasikan browser Anda untuk menggunakan add-on seperti FoxyProxy atau SwitchyOmega.

Untuk informasi selengkapnya tentang cara membuat terowongan SSH, lihat Opsi 2, bagian 1: Siapkan terowongan SSH ke simpul utama menggunakan penerusan port dinamis. Untuk informasi selengkapnya tentang antarmuka web yang tersedia, lihat Melihat antarmuka web yang di-host pada klaster Amazon EMR.

Sertakan pengaturan berikut saat Anda mengatur add-on proksi:

- Gunakan localhost sebagai alamat host.
- Gunakan nomor port lokal yang sama yang Anda pilih untuk membuat terowongan SSH dengan simpul utama. Opsi 2, bagian 1: Siapkan terowongan SSH ke simpul utama menggunakan penerusan port dinamis Sebagai contoh, port *8157*. Port ini juga harus cocok dengan nomor port yang Anda gunakan di PuTTY atau emulator terminal lain yang Anda gunakan untuk menyambungkan.
- Tentukan protokol SOCKS v5. SOCKS v5 memungkinkan Anda mengatur otorisasi pengguna secara opsional.
- Pola URL

  Pola URL berikut harus diizinkan dan ditentukan dengan jenis pola wildcard:

  - *ec2*. *compute*.amazonaws.com* dan *10*.amazonaws.com* pola untuk mencocokkan nama DNS publik cluster di wilayah AS.
  - Pola *ec2*.compute* dan *10*.compute* untuk mencocokkan nama DNS publik klaster di seluruh wilayah lainnya.
  - SEBUAH 10. * pola untuk menyediakan akses ke file JobTracker log di Hadoop. Ubah filter ini jika bertentangan dengan rencana akses jaringan Anda.

- Pola *.ec2.internal* dan *.compute.internal* untuk mencocokkan nama DNS privat (internal) klaster di wilayah `us-east-1` dan seluruh wilayah lain, secara berurutan.

Contoh: Konfigurasi FoxyProxy untuk Firefox

Contoh berikut menunjukkan konfigurasi FoxyProxy Standar (versi 7.5.1) untuk Mozilla Firefox.

FoxyProxy menyediakan satu set alat manajemen proxy. Ini memungkinkan Anda menggunakan server proksi untuk URL yang cocok dengan pola yang sesuai dengan domain yang digunakan oleh instans Amazon EC2 di klaster Amazon EMR Anda.

Untuk menginstal dan mengkonfigurasi FoxyProxy menggunakan Mozilla Firefox

1. Di Firefox, buka https://addons.mozilla.org/, cari FoxyProxy Standar, dan ikuti petunjuk untuk menambahkan FoxyProxy ke Firefox.
2. Menggunakan editor teks, membuat file JSON bernama `foxyproxy-settings.json` dari konfigurasi contoh berikut.

```
{
  "k20d21508277536715": {
    "active": true,
    "address": "localhost",
    "port": 8157,
    "username": "",
    "password": "",
    "type": 3,
    "proxyDNS": true,
    "title": "emr-socks-proxy",
    "color": "#0055E5",
    "index": 9007199254740991,
    "whitePatterns": [
      {
        "title": "*ec2*.*compute*.amazonaws.com*",
        "active": true,
        "pattern": "*ec2*.*compute*.amazonaws.com*",
        "importedPattern": "*ec2*.*compute*.amazonaws.com*",
        "type": 1,
        "protocols": 1
      },
      {
        "title": "*ec2*.compute*",
```

```
        "active": true,
        "pattern": "*ec2*.compute*",
        "importedPattern": "*ec2*.compute*",
        "type": 1,
        "protocols": 1
      },
      {
        "title": "10.*",
        "active": true,
        "pattern": "10.*",
        "importedPattern": "http://10.*",
        "type": 1,
        "protocols": 2
      },
      {
        "title": "*10*.amazonaws.com*",
        "active": true,
        "pattern": "*10*.amazonaws.com*",
        "importedPattern": "*10*.amazonaws.com*",
        "type": 1,
        "protocols": 1
      },
      {
        "title": "*10*.compute*",
        "active": true,
        "pattern": "*10*.compute*",
        "importedPattern": "*10*.compute*",
        "type": 1,
        "protocols": 1
      },
      {
        "title": "*.compute.internal*",
        "active": true,
        "pattern": "*.compute.internal*",
        "importedPattern": "*.compute.internal*",
        "type": 1,
        "protocols": 1
      },
      {
        "title": "*.ec2.internal* ",
        "active": true,
        "pattern": "*.ec2.internal*",
        "importedPattern": "*.ec2.internal*",
        "type": 1,
```

```
        "protocols": 1
      }
    ],
    "blackPatterns": []
  },
  "logging": {
    "size": 100,
    "active": false
  },
  "mode": "patterns",
  "browserVersion": "68.12.0",
  "foxyProxyVersion": "7.5.1",
  "foxyProxyEdition": "standard"
}
```

3. Buka halaman Kelola Ekstensi Anda di Firefox (buka about:addons, lalu pilih Ekstensi).
4. Pilih FoxyProxy Standar, lalu pilih tombol opsi lainnya (tombol yang terlihat seperti elipsis).
5. Pilih Opsi dari menu drop-down.
6. Pilih Pengaturan Impor dari menu sebelah kiri.
7. Pada halaman Pengaturan Impor, pilih Pengaturan Impor di bawah Pengaturan Impor dari FoxyProxy 6.0+, telusuri ke lokasi **`foxyproxy-settings.json`** file yang Anda buat, pilih file, dan pilih Buka.
8. Pilih OKE saat diminta untuk menimpa pengaturan yang ada dan menyimpan konfigurasi baru Anda.

Contoh: Konfigurasi SwitchyOmega untuk chrome

Contoh berikut menunjukkan cara mengatur SwitchyOmega ekstensi untuk Google Chrome. SwitchyOmega memungkinkan Anda mengonfigurasi, mengelola, dan beralih di antara beberapa proxy.

Untuk menginstal dan mengkonfigurasi SwitchyOmega menggunakan Google Chrome

1. Buka https://chrome.google.com/webstore/category/extensions, cari Proxy SwitchyOmega, dan tambahkan ke Chrome.
2. Pilih Profil baru dan masukkan `emr-socks-proxy` sebagai nama profil.
3. Pilih Profil PAC dan kemudian Buat. File Konfigurasi Otomatis Proxy (PAC) membantu Anda menentukan daftar izinkan untuk permintaan peramban yang harus diteruskan ke server proksi web.

4. Dalam bidang Skrip PAC, ganti isinya dengan skrip berikut yang mendefinisikan URL mana yang harus diteruskan melalui server proksi web Anda. Jika Anda menentukan nomor port yang berbeda ketika Anda mengatur terowongan SSH Anda, ganti *8157* dengan nomor port Anda.

```
function FindProxyForURL(url, host) {
    if (shExpMatch(url, "*ec2*.*compute*.amazonaws.com*")) return 'SOCKS5
 localhost:8157';
    if (shExpMatch(url, "*ec2*.compute*")) return 'SOCKS5 localhost:8157';
    if (shExpMatch(url, "http://10.*")) return 'SOCKS5 localhost:8157';
    if (shExpMatch(url, "*10*.compute*")) return 'SOCKS5 localhost:8157';
    if (shExpMatch(url, "*10*.amazonaws.com*")) return 'SOCKS5 localhost:8157';
    if (shExpMatch(url, "*.compute.internal*")) return 'SOCKS5 localhost:8157';
    if (shExpMatch(url, "*ec2.internal*")) return 'SOCKS5 localhost:8157';
    return 'DIRECT';
}
```

5. Di bawah Tindakan, pilih Terapkan perubahan untuk menyimpan setelan proxy Anda.
6. Pada bilah alat Chrome, pilih SwitchyOmega dan pilih `emr-socks-proxy` profil.

Mengakses antarmuka web di peramban

Untuk membuka antarmuka web, masukkan nama DNS publik dari node primer atau inti Anda diikuti dengan nomor port untuk antarmuka yang Anda pilih ke bilah alamat browser Anda. Contoh berikut menunjukkan URL yang akan Anda masukkan untuk terhubung ke Spark HistoryServer.

```
http://master-public-dns-name:18080/
```

Untuk petunjuk tentang mengambil nama DNS publik simpul, lihat Mengambil nama DNS publik dari node utama. Untuk daftar lengkap URL antarmuka web, lihat Melihat antarmuka web yang di-host pada klaster Amazon EMR.

Kirim pekerjaan ke sebuah klaster

Bagian ini menjelaskan metode yang dapat Anda gunakan untuk mengirimkan karya ke klaster EMR Amazon. Untuk mengirimkan pekerjaan, Anda dapat menambahkan langkah-langkah, atau Anda dapat secara interaktif mengirimkan pekerjaan Hadoop ke node utama.

Pertimbangkan aturan perilaku langkah berikut saat Anda mengirimkan langkah ke klaster:

- ID langkah dapat berisi hingga 256 karakter.
- Anda dapat memiliki hingga 256 langkah PENDING dan RUNNING dalam sebuah cluster.
- Bahkan jika Anda memiliki 256 langkah aktif yang berjalan di cluster, Anda dapat secara interaktif mengirimkan pekerjaan ke node utama. Anda dapat mengirimkan jumlah langkah yang tidak terbatas selama usia klaster yang berjalan lama, tetapi hanya 256 langkah dapat BERJALAN atau TERTUNDA pada satu waktu tertentu.
- Dengan Amazon EMR versi 4.8.0 dan yang lebih baru, kecuali versi 5.0.0, Anda dapat membatalkan langkah-langkah yang tertunda. Untuk informasi selengkapnya, lihat Membatalkan langkah.
- Dengan Amazon EMR versi 5.28.0 dan yang lebih baru, Anda dapat membatalkan langkah-langkah tertunda dan berjalan. Anda juga dapat memilih untuk menjalankan beberapa langkah secara paralel untuk meningkatkan pemanfaatan klaster dan menghemat biaya. Untuk informasi selengkapnya, lihat Pertimbangan untuk menjalankan beberapa langkah secara paralel.

> **Note**
>
> Untuk kinerja terbaik, kami menyarankan Anda menyimpan tindakan bootstrap kustom, skrip, dan file lain yang ingin Anda gunakan dengan Amazon EMR di bucket Amazon S3 yang sama dengan cluster Anda. Wilayah AWS

Topik

Menambahkan langkah-langkah ke cluster dengan Amazon EMR Management Console

Gunakan prosedur berikut untuk menambahkan langkah-langkah ke cluster denganAWS Management Console. Untuk informasi terperinci tentang cara mengirimkan langkah-langkah untuk aplikasi big data tertentu, lihat bagian berikut dari Panduan Rilis Amazon EMR:

- Kirim langkah JAR khusus
- Kirim langkah streaming Hadoop
- Kirim langkah Spark
- Kirim langkah Babi
- Jalankan perintah atau skrip sebagai langkah
- Masukkan nilai ke dalam langkah-langkah untuk menjalankan skrip Hive

Tambahkan langkah-langkah selama pembuatan cluster

DariAWS Management Console, Anda dapat menambahkan langkah-langkah saat membuat cluster.

> **Note**
>
> Kami telah mendesain ulang konsol EMR Amazon agar lebih mudah digunakan. Lihat Apa yang baru dengan konsol? untuk mempelajari tentang perbedaan antara pengalaman konsol lama dan baru.

New console

Untuk menambahkan langkah-langkah saat Anda membuat cluster dengan konsol baru

1. Masuk keAWS Management Console, dan buka konsol EMR Amazon di https://console.aws.amazon.com/emr.
2. Di bawah EMR pada EC2 di panel navigasi kiri, pilih Clusters, lalu pilih Create cluster.
3. Di bawah Langkah, pilih Tambahkan langkah. Masukkan nilai yang sesuai di bidang dalam dialog Tambahkan langkah. Untuk informasi tentang memformat argumen langkah Anda, lihatTambahkan argumen langkah. Opsi akan berbeda tergantung pada tipe langkah. Untuk menambahkan langkah Anda dan keluar dari dialog, pilih Tambah langkah.
4. Pilih opsi lain yang berlaku untuk cluster Anda.
5. Untuk meluncurkan klaster Anda, pilih Buat klaster.

Old console

Untuk menambahkan langkah-langkah saat Anda membuat cluster dengan konsol lama

1. Buka konsol EMR Amazon di https://console.aws.amazon.com/elasticmapreduce/home. Pilih Buat Cluster - Opsi Lanjutan.
2. Pada halaman Langkah 1: Perangkat Lunak dan Langkah, untuk Langkah (opsional), pilih Jalankan beberapa langkah secara paralel untuk meningkatkan pemanfaatan klaster dan menghemat biaya. Nilai default untuk tingkat konkurensi adalah 10. Anda dapat memilih antara 2 dan 256 langkah yang dapat berjalan secara paralel.

   > **Note**
   >
   > Menjalankan beberapa langkah secara paralel hanya didukung dengan Amazon EMR versi 5.28.0 dan yang lebih baru.

3. Untuk Setelah langkah terakhir selesai, pilih Klaster memasuki status tunggu atau Akhiri klaster secara otomatis.
4. Pilih Jenis langkah, kemudian Tambahkan Langkah.
5. Ketik nilai yang sesuai di bidang dalam dialog Tambahkan Langkah. Untuk informasi tentang memformat argumen langkah Anda, lihatTambahkan argumen langkah. Opsi akan berbeda tergantung pada tipe langkah. Jika Anda telah mengaktifkan Jalankan beberapa langkah secara paralel untuk meningkatkan pemanfaatan klaster dan menghemat biaya, satu-satunya opsi untuk Action on failure adalah Lanjutkan. Selanjutnya, pilih Tambahkan.

Tambahkan langkah-langkah ke cluster yang sedang berjalan

DenganAWS Management Console, Anda dapat menambahkan langkah-langkah ke cluster dengan opsi penghentian otomatis dinonaktifkan.

New console

Untuk menambahkan langkah-langkah ke cluster yang sedang berjalan dengan konsol baru

1. Masuk keAWS Management Console, dan buka konsol EMR Amazon di https://console.aws.amazon.com/emr.
2. Di bawah EMR pada EC2 di panel navigasi kiri, pilih Cluster, dan pilih cluster yang ingin Anda perbarui.

3. Pada tab Langkah pada halaman detail cluster, pilih Tambah langkah. Untuk mengkloning langkah yang ada, pilih menu dropdown Actions dan pilih Clone step.
4. Masukkan nilai yang sesuai di bidang dalam dialog Tambahkan langkah. Opsi akan berbeda tergantung pada tipe langkah. Untuk menambahkan langkah Anda dan keluar dari dialog, pilih Tambah langkah.

Old console

Untuk menambahkan langkah-langkah ke cluster yang sedang berjalan dengan konsol lama

1. Buka konsol EMR Amazon di https://console.aws.amazon.com/elasticmapreduce/home. Pada halaman Daftar Klaster, pilih tautan untuk klaster Anda.
2. Pada halaman Detail Klaster, pilih tab Langkah.
3. Pada tab Langkah, pilih Tambahkan Langkah.
4. Ketik nilai yang sesuai dalam bidang pada dialog Tambahkan Langkah , lalu pilih Tambahkan. Opsinya berbeda tergantung pada tipe langkah.

Ubah tingkat konkurensi langkah di cluster yang sedang berjalan

DenganAWS Management Console, Anda dapat memodifikasi level konkurensi langkah di cluster yang sedang berjalan.

> **Note**
>
> Anda hanya dapat menjalankan beberapa langkah secara paralel dengan Amazon EMR versi 5.28.0 dan yang lebih baru.

New console

Untuk memodifikasi konkurensi langkah di cluster yang sedang berjalan dengan konsol baru

1. Masuk keAWS Management Console, dan buka konsol EMR Amazon di https://console.aws.amazon.com/emr.
2. Di bawah EMR pada EC2 di panel navigasi kiri, pilih Cluster, dan pilih cluster yang ingin Anda perbarui. Cluster harus berjalan untuk mengubah atribut konkurensi.

3. Pada tab Langkah di halaman detail cluster, temukan bagian Atribut. Pilih Edit untuk mengubah konkurensi. Masukkan nilai antara 1 dan 256.

Old console

Untuk memodifikasi konkurensi langkah di cluster yang sedang berjalan dengan konsol lama

1. Buka konsol EMR Amazon di https://console.aws.amazon.com/elasticmapreduce/home. Pada halaman Daftar Klaster, pilih tautan untuk klaster Anda.
2. Pada halaman Detail Klaster, pilih tab Langkah.
3. Untuk Konkurensi, pilih Ubah. Pilih nilai baru untuk tingkat konkurensi langkah dan kemudian simpan.

Tambahkan argumen langkah

Saat Anda menggunakan AWS Management Console untuk menambahkan langkah ke klaster Anda, Anda dapat menentukan argumen untuk langkah itu di bidang Argumen. Anda harus memisahkan argumen dengan spasi putih dan argumen string surround yang terdiri dari karakter dan spasi dengan tanda kutip.

Example : Argumen yang benar

Contoh argumen berikut diformat dengan benar untukAWS Management Console, dengan tanda kutip di sekitar argumen string akhir.

```
bash -c "aws s3 cp s3://DOC-EXAMPLE-BUCKET/my-script.sh ."
```

Anda juga dapat menempatkan setiap argumen pada baris terpisah untuk keterbacaan seperti yang ditunjukkan pada contoh berikut.

```
bash
-c
"aws s3 cp s3://DOC-EXAMPLE-BUCKET/my-script.sh ."
```

Example : Argumen salah

Contoh argumen berikut tidak diformat dengan benar untuk. AWS Management Console Perhatikan bahwa argumen string akhir,`aws s3 cp s3://DOC-EXAMPLE-BUCKET/my-script.sh .`, berisi spasi dan tidak dikelilingi oleh tanda kutip.

```
bash -c aws s3 cp s3://DOC-EXAMPLE-BUCKET/my-script.sh .
```

Menambahkan langkah-langkah ke cluster dengan AWS CLI

Prosedur berikut menunjukkan cara menambahkan langkah-langkah ke cluster yang baru dibuat dan ke cluster yang sedang berjalan denganAWS CLI. Kedua contoh menggunakan `--steps` subperintah untuk menambahkan langkah-langkah ke cluster.

Untuk menambahkan langkah-langkah selama pembuatan klaster

- Ketik perintah berikut untuk membuat klaster dan menambahkan langkah Apache Pig. Pastikan untuk mengganti *myKey*dengan nama key pair Amazon EC2 Anda.

```
aws emr create-cluster --name "Test cluster" \
--applications Name=Spark \
--use-default-roles \
--ec2-attributes KeyName=myKey \
--instance-groups InstanceGroupType=PRIMARY,InstanceCount=1,InstanceType=m5.xlarge
 InstanceGroupType=CORE,InstanceCount=2,InstanceType=m5.xlarge \
--steps '[{"Args":["spark-submit","--deploy-mode","cluster","--
class","org.apache.spark.examples.SparkPi","/usr/lib/spark/examples/jars/spark-
examples.jar","5"],"Type":"CUSTOM_JAR","ActionOnFailure":"CONTINUE","Jar":"command-
runner.jar","Properties":"","Name":"Spark application"}]'
```

> **Note**
>
> Daftar argumen berubah tergantung pada jenis langkah.

Secara default, tingkat konkurensi langkah adalah 1. Anda dapat mengatur tingkat konkurensi langkah dengan `StepConcurrencyLevel` parameter saat Anda membuat cluster.

Outputnya adalah pengidentifikasi klaster yang serupa dengan berikut ini.

```
{
    "ClusterId": "j-2AXXXXXXGAPLF"
}
```

Untuk menambahkan langkah ke klaster berjalan

- Ketik perintah berikut untuk menambahkan langkah ke klaster berjalan. Ganti *j-2AXXXXXXGAPLF* dengan ID cluster Anda sendiri.

```
aws emr add-steps --cluster-id j-2AXXXXXXGAPLF \
--steps '[{"Args":["spark-submit","--deploy-mode","cluster","--
class","org.apache.spark.examples.SparkPi","/usr/lib/spark/examples/jars/spark-
examples.jar","5"],"Type":"CUSTOM_JAR","ActionOnFailure":"CONTINUE","Jar":"command-
runner.jar","Properties":"","Name":"Spark application"}]'
```

  Outputnya adalah pengidentifikasi langkah yang serupa dengan berikut ini.

```
{
    "StepIds": [
 "s-Y9XXXXXXAPMD"
    ]
}
```

Untuk memodifikasi StepConcurrencyLevel dalam cluster yang sedang berjalan

1. Di cluster yang sedang berjalan, Anda dapat memodifikasi `StepConcurrencyLevel` dengan `ModifyCluster` API. Misalnya, ketik perintah berikut untuk meningkatkan `StepConcurrencyLevel` ke`10`. Ganti *j-2AXXXXXXGAPLF* dengan ID cluster Anda.

```
aws emr modify-cluster --cluster-id j-2AXXXXXXGAPLF --step-concurrency-level 10
```

2. Output Anda serupa dengan yang berikut ini.

```
{
"StepConcurrencyLevel": 10
}
```

Untuk informasi selengkapnya tentang penggunaan perintah EMR Amazon diAWS CLI, lihat Referensi AWS CLIPerintah.

Pertimbangan untuk menjalankan beberapa langkah secara paralel

- Langkah-langkah yang berjalan secara paralel dapat diselesaikan dalam urutan apa pun, tetapi langkah-langkah tertunda dalam antrean akan bertransisi ke keadaan berjalan sesuai urutan dikirimkan.
- Ketika Anda memilih tingkat konkurensi langkah untuk klaster Anda, Anda harus mempertimbangkan apakah jenis instance node utama memenuhi persyaratan memori beban kerja pengguna atau tidak. Proses eksekusi langkah utama berjalan pada node utama untuk setiap langkah. Menjalankan beberapa langkah secara paralel membutuhkan lebih banyak memori dan pemanfaatan CPU dari node utama daripada menjalankan satu langkah pada satu waktu.
- Untuk mencapai penjadwalan yang kompleks dan pengelolaan sumber daya dari langkah-langkah bersamaan, Anda dapat menggunakan fitur penjadwalan YARN seperti `FairScheduler` atau `CapacityScheduler`. Misalnya, Anda dapat menggunakan `FairScheduler` dengan `queueMaxAppsDefault` diatur untuk mencegah lebih dari sejumlah pekerjaan berjalan pada satu waktu.
- Tingkat konkurensi langkah tunduk pada konfigurasi pengelola sumber daya. Sebagai contoh, jika YARN dikonfigurasi dengan hanya paralelisme `5`, maka Anda hanya dapat memiliki lima aplikasi YARN yang berjalan secara paralel bahkan jika `StepConcurrencyLevel` diatur ke `10`. Untuk informasi selengkapnya tentang mengonfigurasi pengelola sumber daya, lihat Mengonfigurasi aplikasi di Panduan Rilis EMR Amazon.
- Anda tidak dapat menambahkan langkah dengan `ActionOnFailure` selain LANJUTKAN jika tingkat konkurensi langkah klaster lebih besar dari 1.
- Jika tingkat konkurensi langkah klaster lebih besar dari satu, fitur langkah `ActionOnFailure` tidak akan teraktivasi.
- Jika sebuah klaster memiliki tingkat konkurensi langkah 1 tetapi memiliki beberapa langkah berjalan, `TERMINATE_CLUSTER ActionOnFailure` dapat teraktivasi, tetapi `CANCEL_AND_WAIT ActionOnFailure` tidak. Kasus edge ini muncul ketika tingkat konkurensi langkah klaster lebih besar dari satu, tapi akan turun jika ada beberapa langkah berjalan.
- Anda dapat menggunakan penskalaan otomatis EMR untuk menaikkan skala dan menurunkan skala berdasarkan sumber daya YARN guna mencegah perebutan sumber daya. Untuk informasi selengkapnya, lihat Menggunakan penskalaan otomatis dengan kebijakan khusus untuk grup instans di Panduan Manajemen EMR Amazon.
- Ketika Anda menurunkan tingkat konkurensi langkah, EMR mengizinkan langkah-langkah berjalan untuk diselesaikan sebelum mengurangi jumlah langkah. Jika sumber daya habis karena klaster

menjalankan terlalu banyak langkah secara bersamaan, kami merekomendasikan untuk secara manual membatalkan langkah-langkah berjalan untuk mengosongkan sumber daya.

Melihat langkah-langkah

Total jumlah catatan langkah yang dapat Anda lihat (terlepas dari statusnya) adalah 1.000. Total ini mencakup langkah-langkah yang dikirim pengguna dan sistem. Ketika status langkah-langkah yang dikirim pengguna berubah ke SELESAI atau GAGAL, langkah-langkah tambahan yang dikirim pengguna dapat ditambahkan ke klaster sampai batas langkah 1.000 tercapai. Setelah 1.000 langkah telah ditambahkan ke klaster, pengiriman langkah-langkah tambahan tersebut menyebabkan penghapusan catatan langkah yang dikirim pengguna yang lama. Catatan-Catatan ini tidak dihapus dari berkas log. Mereka akan dihapus dari tampilan konsol, dan mereka tidak muncul ketika Anda menggunakan AWS CLI atau API untuk mengambil informasi klaster. Catatan langkah sistem tidak pernah dihapus.

Informasi langkah yang dapat Anda lihat tergantung pada mekanisme yang digunakan untuk mengambil informasi klaster. Tabel berikut menunjukkan informasi langkah yang dikembalikan oleh masing-masing pilihan yang tersedia.

Opsi	DescribeJobFlow atau --describe --jobflow	ListSteps atau daftar langkah
SDK	256 langkah	1.000 langkah
CLI Amazon EMR	256 langkah	NA
AWS CLI	NA	1.000 langkah
API	256 langkah	1.000 langkah

Membatalkan langkah

Anda dapat membatalkan langkah-langkah yang tertunda dan berjalan dariAWS Management Console, API EMR AmazonAWS CLI, atau Amazon.

> **Note**
>
> Kami telah mendesain ulang konsol EMR Amazon agar lebih mudah digunakan. Lihat Apa yang baru dengan konsol? untuk mempelajari tentang perbedaan antara pengalaman konsol lama dan baru.

New console

Untuk membatalkan langkah-langkah dengan konsol baru

1. Masuk keAWS Management Console, dan buka konsol EMR Amazon di https://console.aws.amazon.com/emr.
2. Di bawah EMR pada EC2 di panel navigasi kiri, pilih Cluster, lalu pilih cluster yang ingin Anda perbarui.
3. Pada tab Langkah pada halaman detail klaster, pilih kotak centang di sebelah langkah yang ingin Anda batalkan. Pilih menu tarik-turun Tindakan dan kemudian pilih Batalkan langkah.
4. Dalam dialog Batalkan langkah, pilih untuk membatalkan langkah dan tunggu sampai keluar, atau batalkan langkah dan paksa untuk keluar. Kemudian pilih Konfirmasi.
5. Status langkah-langkah dalam tabel Langkah berubah menjadi`CANCELLED`.

Old console

Untuk membatalkan langkah-langkah dengan konsol lama

1. Arahkan ke konsol EMR Amazon baru dan pilih Beralih ke konsol lama dari navigasi samping. Untuk informasi selengkapnya tentang apa yang diharapkan saat beralih ke konsol lama, lihat Menggunakan konsol lama.
2. Pada halaman Detail Klaster, perluas bagian Langkah.
3. Untuk setiap langkah yang ingin Anda batalkan, pilih langkah dari daftar Langkah. Kemudian pilih Batalkan langkah.
4. Di dialog Batalkan langkah, biarkan opsi default Batalkan langkah dan tunggu sampai keluar. Jika Anda ingin segera mengakhiri langkah tanpa menunggu proses selesai, pilih Batalkan langkah dan paksa untuk keluar.
5. Kemudian pilih Batalkan langkah.

CLI

Untuk membatalkan dengan menggunakan AWS CLI

- Gunakan perintah `aws emr cancel-steps`, tentukan klaster dan langkah-langkah untuk dibatalkan. Contoh berikut menunjukkan perintah AWS CLI untuk membatalkan dua langkah.

```
aws emr cancel-steps --cluster-id j-2QUAXXXXXXXXX \
--step-ids s-3M8DXXXXXXXXX s-3M8DXXXXXXXXX \
--step-cancellation-option SEND_INTERRUPT
```

Dengan Amazon EMR versi 5.28.0, Anda dapat memilih salah satu dari dua opsi pembatalan berikut untuk parameter `StepCancellationOption` saat membatalkan langkah.

- `SEND_INTERRUPT`— Ini adalah pilihan default. Saat permintaan pembatalan langkah diterima, EMR akan mengirimkan sinyal `SIGTERM` ke langkah tersebut. Tambahkan penangan sinyal `SIGTERM` ke logika langkah Anda untuk menangkap sinyal ini dan mengakhiri proses langkah turunan atau menunggu mereka selesai.
- `TERMINATE_PROCESS` — Ketika opsi ini dipilih, EMR mengirimkan sinyal `SIGKILL` ke langkah dan semua proses turunannya guna mengakhiri mereka segera.

Pertimbangan untuk membatalkan langkah-langkah

- Membatalkan langkah yang berjalan atau tertunda akan menghapus langkah tersebut dari jumlah langkah aktif.
- Membatalkan langkah berjalan tidak akan mengizinkan langkah tertunda untuk mulai berjalan, dengan asumsi tidak ada perubahan ke `stepConcurrencyLevel`.
- Membatalkan langkah berjalan tidak memicu langkah `ActionOnFailure`.
- Untuk EMR 5.32.0 dan yang lebih baru, `SEND_INTERRUPT StepCancellationOption` mengirimkan sinyal `SIGTERM` untuk proses anak langkah tersebut. Anda harus memperhatikan sinyal ini dan melakukan pembersihan dan shutdown secara perlahan. `TERMINATE_PROCESS StepCancellationOption` mengirimkan sinyal `SIGKILL` untuk proses anak langkah dan semua proses turunannya; Namun, proses asinkron tidak terpengaruh.

Melihat dan memantau suatu klaster

Amazon EMR menyediakan beberapa alat yang dapat Anda gunakan untuk mengumpulkan informasi tentang klaster Anda. Anda dapat mengakses informasi tentang klaster dari konsol, CLI atau secara terprogram. Antarmuka web Hadoop standar dan file log tersedia di simpul utama. Anda juga dapat menggunakan layanan pemantauan seperti CloudWatch dan Ganglia untuk melacak kinerja cluster Anda.

Riwayat aplikasi juga tersedia dari konsol menggunakan UI aplikasi “persisten” untuk Server Riwayat Spark mulai dengan Amazon EMR 5.25.0. Dengan Amazon EMR 6.x, server timeline YARN persisten, dan antarmuka pengguna Tez juga tersedia. Layanan ini di-host di luar klaster, sehingga Anda dapat mengakses riwayat aplikasi selama 30 hari setelah klaster berakhir, tanpa perlu untuk koneksi SSH atau proksi web. Lihat Melihat riwayat aplikasi.

Topik

Melihat status dan detail klaster

Setelah Anda membuat sebuah klaster, Anda dapat memantau statusnya dan mendapatkan informasi detail tentang eksekusi dan kesalahan yang mungkin terjadi, bahkan setelah klaster tersebut diakhiri. Amazon EMR menyimpan metadata tentang klaster yang diakhiri untuk referensi Anda selama dua bulan, setelah metadata dihapus. Anda tidak dapat menghapus klaster dari riwayat klaster, tetapi menggunakan AWS Management Console, Anda dapat menggunakan Filter, dan menggunakan AWS CLI, Anda dapat menggunakan opsi dengan perintah `list-clusters` untuk fokus pada klaster yang Anda pedulikan.

Anda dapat mengakses riwayat aplikasi yang disimpan di klaster selama satu minggu dari waktu riwayat tersebut dicatat, terlepas dari apakah klaster tersebut berjalan atau diakhiri. Selain itu,

antarmuka pengguna aplikasi persisten menyimpan riwayat aplikasi di luar klaster selama 30 hari setelah klaster berakhir. Lihat Melihat riwayat aplikasi.

Untuk informasi selengkapnya tentang status klaster, seperti Menunggu dan Berjalan, lihat Memahami siklus hidup klaster.

Lihat detail klaster menggunakan AWS Management Console

Daftar Cluster di https://console.aws.amazon.com/emr mencantumkan semua cluster di akun dan AWS Wilayah Anda, termasuk cluster yang dihentikan. Daftar ini menunjukkan hal berikut untuk setiap cluster: Nama dan ID, detail Status dan Status, waktu Pembuatan, waktu Berlalu saat cluster berjalan, dan jam instans Normalisasi yang telah diperoleh untuk semua instans EC2 di cluster. Daftar ini adalah titik mulai untuk memantau status klaster Anda. Ini dirancang agar Anda dapat menelusuri detail setiap klaster untuk analisis dan pemecahan masalah.

> **Note**
>
> Kami telah mendesain ulang konsol EMR Amazon agar lebih mudah digunakan. Lihat Apa yang baru dengan konsol? untuk mempelajari tentang perbedaan antara pengalaman konsol lama dan baru.

New console

Untuk melihat informasi cluster dengan konsol baru

1. Masuk keAWS Management Console, dan buka konsol EMR Amazon di https://console.aws.amazon.com/emr.
2. Di bawah EMR pada EC2 di panel navigasi kiri, pilih Clusters, dan pilih cluster yang ingin Anda lihat.
3. Gunakan panel Ringkasan untuk melihat dasar-dasar konfigurasi klaster Anda, seperti status klaster, aplikasi sumber terbuka yang diinstal Amazon EMR di klaster, dan versi Amazon EMR yang Anda gunakan untuk membuat klaster. Gunakan setiap tab di bawah Ringkasan untuk melihat informasi seperti yang dijelaskan dalam tabel berikut.

Old console

Untuk melihat informasi cluster dengan konsol lama

1. Arahkan ke konsol EMR Amazon baru dan pilih Beralih ke konsol lama dari navigasi samping. Untuk informasi selengkapnya tentang apa yang diharapkan saat beralih ke konsol lama, lihat Menggunakan konsol lama.
2. Untuk melihat ringkasan singkat informasi klaster, pilih panah bawah di sebelah tautan untuk cluster di bawah Nama. Baris klaster diperluas untuk memberikan informasi lebih lanjut tentang klaster tersebut, perangkat keras, langkah-langkah, dan tindakan bootstrap. Gunakan tautan di bagian ini untuk menggali hal-hal spesifik. Misalnya, klik tautan di bawah Langkah-Langkah untuk mengakses berkas log langkah, melihat JAR terkait langkah tersebut, mendalami pekerjaan dan tugas dalam langkah tersebut, dan mengakses berkas log.
3. Untuk melihat informasi klaster secara mendalam, pilih tautan cluster di bawah Nama untuk membuka halaman detail cluster. Informasi berikut tersedia di halaman detail cluster di konsol lama:

Tab (Konsol lama)	Deskripsi (Konsol lama)
Sifat-sifat	Gunakan tab ini untuk melihat sistem operasi klaster Anda, penghentian klaster dan konfigurasi keamanan, informasi VPC dan subnet Anda, dan tempat Anda menyimpan log di Amazon S3.
Tindakan Bootstrap	Gunakan tab ini untuk melihat status dari setiap tindakan bootstrap yang dijalanka n klaster ketika ia diluncurkan. Tindakan bootstrap digunakan untuk instalasi perangkat lunak kustom dan konfigurasi lanjutan. Untuk informasi selengkapnya, lihat Buat tindakan bootstrap untuk menginstal perangkat lunak tambahan.
Pemantauan	Gunakan tab ini untuk melihat metrik utama operasi klaster. Anda dapat melihat data

Tab (Konsol lama)	Deskripsi (Konsol lama)
	tingkat klaster, data tingkat simpul, dan informasi tentang I/O dan penyimpanan data.
Contoh	Gunakan tab ini untuk melihat informasi tentang node di klaster Anda, termasuk ID instans EC2, nama DNS, volume EBS, dan lainnya.
Langkah-langkah	Gunakan tab ini untuk melihat status dan mengakses berkas log untuk langkah-l angkah yang Anda kirimkan. Untuk informasi selengkapnya tentang langkah-langkahnya, lihat Kirim pekerjaan ke sebuah klaster.
Aplikasi	Gunakan tab ini untuk melihat server timeline YARN luar klaster persisten dan detail aplikasi Tez UI. Anda juga dapat melihat informasi tentang aplikasi yang diinstal, konfigurasi klaster, dan grup instans. Antarmuka pengguna aplikasi di klaster tersedia saat klaster berjalan.
Kejadian	Gunakan tab ini untuk melihat log peristiwa untuk klaster Anda. Untuk informasi selengkapnya, lihat Memantau peristiwa EMR Amazon dengan CloudWatch.
Tanda	Gunakan tab ini untuk melihat tag apa pun yang Anda terapkan ke cluster.

Lihat detail klaster menggunakan AWS CLI

Contoh-contoh berikut ini mendemonstrasikan cara mengambil detail klaster menggunakan AWS CLI. Untuk informasi selengkapnya tentang perintah yang tersedia, lihat AWS CLI Referensi Perintah untuk Amazon EMR. Anda dapat menggunakan perintah describe-cluster untuk melihat detail tingkat klaster termasuk status, konfigurasi perangkat keras dan perangkat lunak, pengaturan VPC, tindakan

bootstrap, grup instans, dan sebagainya. Untuk informasi selengkapnyua tentang status klaster, lihat Memahami siklus hidup klaster. Contoh berikut menunjukkan menggunakan perintah `describe-cluster`, diikuti oleh contoh-contoh perintah list-clusters.

Example Melihat status klaster

Untuk menggunakan perintah `describe-cluster`, Anda memerlukan ID klaster. Contoh ini menunjukkan menggunakan untuk mendapatkan daftar klaster yang dibuat dalam kisaran tanggal tertentu, dan kemudian menggunakan salah satu ID klaster yang dikembalikan untuk mencantumkan informasi selengkapnya tentang status klaster individu.

Perintah berikut menggambarkan klaster *`j-1K48XXXXXXHCB`*, yang Anda ganti dengan ID klaster Anda.

```
aws emr describe-cluster --cluster-id j-1K48XXXXXXHCB
```

Output perintah Anda serupa dengan yang berikut ini:

```
{
    "Cluster": {
        "Status": {
            "Timeline": {
                "ReadyDateTime": 1438281058.061,
                "CreationDateTime": 1438280702.498
            },
            "State": "WAITING",
            "StateChangeReason": {
                "Message": "Waiting for steps to run"
            }
        },
        "Ec2InstanceAttributes": {
            "EmrManagedMasterSecurityGroup": "sg-cXXXXX0",
            "IamInstanceProfile": "EMR_EC2_DefaultRole",
            "Ec2KeyName": "myKey",
            "Ec2AvailabilityZone": "us-east-1c",
            "EmrManagedSlaveSecurityGroup": "sg-example"
        },
        "Name": "Development Cluster",
        "ServiceRole": "EMR_DefaultRole",
        "Tags": [],
        "TerminationProtected": false,
        "ReleaseLabel": "emr-4.0.0",
```

```
        "NormalizedInstanceHours": 16,
        "InstanceGroups": [
            {
                "RequestedInstanceCount": 1,
                "Status": {
                    "Timeline": {
                        "ReadyDateTime": 1438281058.101,
                        "CreationDateTime": 1438280702.499
                    },
                    "State": "RUNNING",
                    "StateChangeReason": {
                        "Message": ""
                    }
                },
                "Name": "CORE",
                "InstanceGroupType": "CORE",
                "Id": "ig-2EEXAMPLEXXP",
                "Configurations": [],
                "InstanceType": "m5.xlarge",
                "Market": "ON_DEMAND",
                "RunningInstanceCount": 1
            },
            {
                "RequestedInstanceCount": 1,
                "Status": {
                    "Timeline": {
                        "ReadyDateTime": 1438281023.879,
                        "CreationDateTime": 1438280702.499
                    },
                    "State": "RUNNING",
                    "StateChangeReason": {
                        "Message": ""
                    }
                },
                "Name": "MASTER",
                "InstanceGroupType": "MASTER",
                "Id": "ig-2A1234567XP",
                "Configurations": [],
                "InstanceType": "m5.xlarge",
                "Market": "ON_DEMAND",
                "RunningInstanceCount": 1
            }
        ],
        "Applications": [
```

```
            {
                "Version": "1.0.0",
                "Name": "Hive"
            },
            {
                "Version": "2.6.0",
                "Name": "Hadoop"
            },
            {
                "Version": "0.14.0",
                "Name": "Pig"
            },
            {
                "Version": "1.4.1",
                "Name": "Spark"
            }
        ],
        "BootstrapActions": [],
        "MasterPublicDnsName": "ec2-X-X-X-X.compute-1.amazonaws.com",
        "AutoTerminate": false,
        "Id": "j-jobFlowID",
        "Configurations": [
            {
                "Properties": {
                    "hadoop.security.groups.cache.secs": "250"
                },
                "Classification": "core-site"
            },
            {
                "Properties": {
                    "mapreduce.tasktracker.reduce.tasks.maximum": "5",
                    "mapred.tasktracker.map.tasks.maximum": "2",
                    "mapreduce.map.sort.spill.percent": "90"
                },
                "Classification": "mapred-site"
            },
            {
                "Properties": {
                    "hive.join.emit.interval": "1000",
                    "hive.merge.mapfiles": "true"
                },
                "Classification": "hive-site"
            }
        ]
```

```
    }
}
```

Example Mencantumkan klaster berdasarkan tanggal pembuatan

Untuk mengambil klaster yang dibuat dalam kisaran data tertentu, gunakan perintah `list-clusters` dengan parameter `--created-after` dan `--created-before`.

Perintah berikut mencantumkan semua klaster yang dibuat antara 09 Oktober 2019 dan 12 Oktober 2019.

```
aws emr list-clusters --created-after 2019-10-09T00:12:00 --created-
before 2019-10-12T00:12:00
```

Example Mencantumkan klaster berdasarkan status

Untuk mencantumkan klaster berdasarkan status, gunakan perintah `list-clusters` dengan parameter `--cluster-states`. Status klaster yang valid meliputi: MULAI, BOOTSTRAPPING, BERJALAN, MENUNGGU, MENGAKHIRI, DIAKHIRI, dan DIAKHIRI_DENGAN_KESALAHAN.

```
aws emr list-clusters --cluster-states TERMINATED
```

Anda juga dapat menggunakan parameter jalan pintas berikut untuk mencantumkan semua klaster dalam status yang ditentukan. :

- `--active` mem-filter klaster dalam status MULAI, BOOTSTRAPPING, BERJALAN, MENUNGGU, atau MENGAKHIRI.
- `--terminated` mem-filter klaster dalam status DIAKHIRI.
- Parameter `--failed` mem-filter klaster dalam status DIAKHIRI_DENGAN_KESALAHAN.

Perintah berikut mengembalikan hasil yang sama.

```
aws emr list-clusters --cluster-states TERMINATED
```

```
aws emr list-clusters --terminated
```

Untuk informasi selengkapnyua tentang status klaster, lihat Memahami siklus hidup klaster.

Debug langkah yang disempurnakan

Jika langkah Amazon EMR gagal dan Anda mengirimkan pekerjaan Anda menggunakan operasi API langkah dengan AMI versi 5.x atau yang lebih baru, Amazon EMR dapat mengidentifikasi dan mengembalikan akar masalah kegagalan langkah dalam beberapa kasus, bersama dengan nama berkas log yang relevan dan sebagian dari jejak tumpukan aplikasi melalui API. Misalnya, kegagalan berikut dapat diidentifikasi:

- Kesalahan Hadoop umum seperti direktori output sudah ada, direktori input tidak ada, atau aplikasi kehabisan memori.
- Kesalahan Java seperti aplikasi yang dikompilasi dengan versi Java yang tidak kompatibel atau dijalankan dengan kelas utama yang tidak ditemukan.
- Masalah mengakses objek yang disimpan di Amazon S3.

Informasi ini tersedia menggunakan operasi DescribeStepdan ListStepsAPI. FailureDetailsBidang yang StepSummarydikembalikan oleh operasi tersebut. Untuk mengakses FailureDetails informasi, gunakan AWS CLI, konsol, atau AWS SDK.

> **Note**
>
> Kami telah mendesain ulang konsol EMR Amazon agar lebih mudah digunakan. Lihat Apa yang baru dengan konsol? untuk mempelajari tentang perbedaan antara pengalaman konsol lama dan baru.

New console

Konsol EMR Amazon baru tidak menawarkan langkah debugging. Namun, Anda dapat melihat detail terminasi cluster dengan langkah-langkah berikut.

Untuk melihat detail kegagalan dengan konsol baru

1. Masuk keAWS Management Console, dan buka konsol EMR Amazon di https://console.aws.amazon.com/emr.
2. Di bawah EMR pada EC2 di panel navigasi kiri, pilih Clusters, lalu pilih cluster yang ingin Anda lihat.
3. Perhatikan nilai Status di bagian Ringkasan halaman detail cluster. Jika status Diakhiri dengan kesalahan, arahkan kursor ke teks untuk melihat detail kegagalan klaster.

Old console

Untuk melihat detail kegagalan dengan konsol lama

1. Arahkan ke konsol EMR Amazon baru dan pilih Beralih ke konsol lama dari navigasi samping. Untuk informasi selengkapnya tentang apa yang diharapkan saat beralih ke konsol lama, lihat Menggunakan konsol lama.
2. Pilih Daftar Klaster dan pilih sebuah klaster.
3. Pilih ikon panah di samping setiap langkah untuk melihat detail lainnya. Jika langkah telah gagal dan Amazon EMR dapat mengidentifikasi akar masalah, Anda melihat detail kegagalan.

CLI

Untuk melihat detail kegagalan dengan AWS CLI

- Untuk mendapatkan detail kegagalan untuk langkah denganAWS CLI, gunakan `describe-step` perintah.

```
aws emr describe-step --cluster-id j-1K48XXXXXHCB --step-id s-3QM0XXXXXM1W
```

Output akan terlihat serupa dengan yang berikut ini:

```
{
  "Step": {
    "Status": {
      "FailureDetails": {
        "LogFile": "s3://myBucket/logs/j-1K48XXXXXHCB/steps/s-3QM0XXXXXM1W/
stderr.gz",
        "Message": "org.apache.hadoop.mapred.FileAlreadyExistsException: Output
 directory s3://myBucket/logs/beta already exists",
        "Reason": "Output directory already exists."
      },
      "Timeline": {
        "EndDateTime": 1469034209.143,
        "CreationDateTime": 1469033847.105,
        "StartDateTime": 1469034202.881
      },
      "State": "FAILED",
      "StateChangeReason": {}
```

```
    },
    "Config": {
      "Args": [
        "wordcount",
        "s3://myBucket/input/input.txt",
        "s3://myBucket/logs/beta"
      ],
      "Jar": "s3://myBucket/jars/hadoop-mapreduce-examples-2.7.2-amzn-1.jar",
      "Properties": {}
    },
    "Id": "s-3QM0XXXXXM1W",
    "ActionOnFailure": "CONTINUE",
    "Name": "ExampleJob"
  }
}
```

Melihat riwayat aplikasi

Anda dapat melihat detail aplikasi layanan layanan timeline Spark History Server dan YARN dengan halaman detail cluster di konsol. Riwayat aplikasi Amazon EMR mamudahkan Anda untuk memecahkan masalah dan menganalisis pekerjaan aktif dan riwayat pekerjaan.

> **Note**
>
> Untuk meningkatkan keamanan aplikasi off-console yang mungkin Anda gunakan dengan Amazon EMR, domain hosting aplikasi terdaftar di Daftar Akhiran Publik (PSL). Contoh domain hosting ini meliputi:`emrstudio-prod.us-east-1.amazonaws.com`,`emrnotebooks-prod.us-east-1.amazonaws.com`,`emrappui-prod.us-east-1.amazonaws.com`. Untuk keamanan lebih lanjut, jika Anda perlu mengatur cookie sensitif di nama domain default, kami sarankan Anda menggunakan cookie dengan `__Host-` awalan. Ini membantu mempertahankan domain Anda dari upaya pemalsuan permintaan lintas situs (CSRF). Untuk informasi selengkapnya, lihat Set-Cookiehalaman di Jaringan Pengembang Mozilla.

Bagian antarmuka pengguna Aplikasi pada tab Aplikasi menyediakan beberapa opsi tampilan, tergantung pada status cluster dan aplikasi yang Anda instal di cluster.

- Akses off-cluster ke antarmuka pengguna aplikasi persisten - Dimulai dengan Amazon EMR versi 5.25.0, tautan antarmuka pengguna aplikasi persisten tersedia untuk Spark UI dan Spark History Service. Dengan Amazon EMR versi 5.30.1 dan yang lebih baru, Tez UI dan server timeline YARN juga memiliki antarmuka pengguna aplikasi yang persisten. Server timeline YARN dan Tez UI adalah aplikasi sumber terbuka yang menyediakan metrik untuk klaster aktif dan diakhiri. Antarmuka pengguna Spark memberikan detail tentang tahapan penjadwal dan tugas, ukuran RDD dan penggunaan memori, informasi lingkungan, dan informasi tentang pelaksana yang sedang berjalan. UI aplikasi persisten dijalankan di luar klaster, sehingga informasi dan log klaster tersedia selama 30 hari setelah aplikasi berakhir. Tidak seperti antarmuka pengguna aplikasi di klaster, UI aplikasi persisten tidak mengharuskan Anda untuk mengatur proksi web melalui koneksi SSH.
- Antarmuka pengguna aplikasi di klaster — Ada berbagai antarmuka pengguna riwayat aplikasi yang dapat dijalankan pada sebuah klaster. Antarmuka pengguna di klaster di-host pada simpul utama dan mengharuskan Anda untuk mengatur koneksi SSH ke server web. Antarmuka pengguna aplikasi di klaster menyimpan riwayat aplikasi selama satu minggu setelah aplikasi berakhir. Untuk informasi selengkapnya dan petunjuk tentang pengaturan terowongan SSH, lihat Melihat antarmuka web yang di-host pada klaster Amazon EMR.

  Dengan pengecualian dari Server Riwayat Spark, server timeline YARN, dan aplikasi Hive, riwayat aplikasi di klaster hanya dapat dilihat saat klaster berjalan.

Melihat antarmuka pengguna aplikasi persisten

Dimulai dengan Amazon EMR versi 5.25.0, Anda dapat terhubung ke detail aplikasi Server Riwayat Spark persisten yang di-host di luar klaster menggunakan halaman Ringkasan klaster atau tab Antarmuka pengguna aplikasi di konsol tersebut. Tez UI dan antarmuka aplikasi persisten server timeline YARN tersedia mulai dari Amazon EMR versi 5.30.1. Akses tautan satu klik ke riwayat aplikasi persisten memberikan manfaat berikut:

- Anda dapat dengan cepat menganalisis dan memecahkan masalah pekerjaan yang aktif dan riwayat pekerjaan tanpa mengatur proksi web melalui koneksi SSH.
- Anda dapat mengakses riwayat aplikasi dan berkas log yang relevan untuk klaster yang aktif dan diakhiri. Log tersedia selama 30 hari setelah aplikasi berakhir.

Arahkan ke detail klaster Anda di konsol, dan pilih tab Aplikasi. Pilih UI aplikasi yang Anda inginkan setelah cluster Anda diluncurkan. UI aplikasi terbuka di tab browser baru. Untuk informasi selengkapnya, lihat Pemantauan dan instrumentasi.

Anda dapat melihat log kontainer YARN melalui tautan pada server riwayat Spark, server timeline YARN, dan Tez UI.

> **Note**
>
> Untuk mengakses log kontainer YARN dari server riwayat Spark, server timeline YARN, dan Tez UI, Anda harus mengaktifkan logging ke Amazon S3 untuk klaster Anda. Jika Anda tidak mengaktifkan logging, tautan ke log kontainer YARN tidak akan berfungsi.

Pengumpulan log

Untuk mengaktifkan akses satu klik ke antarmuka pengguna aplikasi persisten, Amazon EMR mengumpulkan dua jenis log:

- Log peristiwa aplikasi dikumpulkan ke dalam bucket sistem EMR. Log peristiwa dienkripsi saat istirahat menggunakan Enkripsi Sisi Server dengan Kunci Terkelola Amazon S3 (SSE-S3). Jika Anda menggunakan subnet privat untuk klaster Anda, pastikan untuk menyertakan "`arn:aws:s3:::prod.MyRegion.appinfo.src/*`" dalam daftar sumber daya kebijakan Amazon S3 untuk subnet privat. Untuk informasi selengkapnya, lihat Kebijakan Amazon S3 minimum untuk subnet privat.
- Log kontainer YARN dikumpulkan ke dalam bucket Amazon S3 yang Anda miliki. Anda harus mengaktifkan logging untuk klaster Anda untuk mengakses log kontainer YARN. Untuk informasi selengkapnya, lihat Mengkonfigurasi logging dan debug klaster.

Jika Anda perlu untuk menonaktifkan fitur ini untuk alasan privasi, Anda dapat menghentikan daemon dengan menggunakan skrip bootstrap ketika Anda membuat sebuah klaster, seperti yang ditunjukkan contoh berikut.

```
aws emr create-cluster --name "Stop Application UI Support" --release-label emr-7.0.0 \
--applications Name=Hadoop Name=Spark --ec2-attributes KeyName=<myEMRKeyPairName> \
--instance-groups InstanceGroupType=MASTER,InstanceCount=1,InstanceType=m3.xlarge
 InstanceGroupType=CORE,InstanceCount=1,InstanceType=m3.xlarge
 InstanceGroupType=TASK,InstanceCount=1,InstanceType=m3.xlarge \
--use-default-roles --bootstrap-actions Path=s3://region.elasticmapreduce/bootstrap-
actions/run-if,Args=["instance.isMaster=true","echo Stop Application UI | sudo tee /
etc/apppusher/run-apppusher; sudo systemctl stop apppusher || exit 0"]
```

Setelah Anda menjalankan skrip bootstrap ini, Amazon EMR tidak akan mengumpulkan log peristiwa Server Riwayat Spark atau server timeline YARN ke bucket sistem EMR. Tidak ada informasi riwayat aplikasi yang akan tersedia di tab Antarmuka pengguna aplikasi, dan Anda akan kehilangan akses ke semua antarmuka pengguna aplikasi dari konsol tersebut.

File log peristiwa Spark besar

Dalam beberapa kasus, pekerjaan Spark yang berjalan lama, seperti streaming Spark, dan pekerjaan besar, seperti kueri Spark SQL, dapat menghasilkan log peristiwa besar. Dengan log peristiwa besar, Anda dapat dengan cepat menggunakan ruang disk pada instance komputasi dan mengalami `OutOfMemory` kesalahan saat memuat UI Persisten. Untuk menghindari masalah ini, kami sarankan Anda mengaktifkan fitur penggulungan dan pemadatan log peristiwa Spark. Fitur ini tersedia di Amazon EMR versi emr-6.1.0 dan yang lebih baru. Untuk detail selengkapnya tentang rolling dan compaction, lihat Menerapkan pemadatan pada file log peristiwa bergulir dalam dokumentasi Spark.

Untuk mengaktifkan fitur penggulungan dan pemadatan log peristiwa Spark, aktifkan pengaturan konfigurasi Spark berikut.

- `spark.eventLog.rolling.enabled`— Menghidupkan log acara bergulir berdasarkan ukuran. Pengaturan ini dinonaktifkan secara default.
- `spark.eventLog.rolling.maxFileSize`— Saat penggulungan diaktifkan, tentukan ukuran maksimum file log peristiwa sebelum berguling. Defaultnya adalah 128 MB.
- `spark.history.fs.eventLog.rolling.maxFilesToRetain`- Menentukan jumlah maksimum file log peristiwa non-dipadatkan untuk mempertahankan. Secara default, semua file log peristiwa dipertahankan. Setel ke angka yang lebih rendah untuk memadatkan log peristiwa lama. Nilai terendah adalah 1.

Perhatikan bahwa pemadatan mencoba untuk mengecualikan peristiwa dengan file log peristiwa yang sudah ketinggalan zaman, seperti berikut ini. Jika tidak membuang peristiwa, Anda tidak lagi melihatnya di UI Server Riwayat Spark.

- Acara untuk pekerjaan jadi dan acara panggung atau tugas terkait.
- Acara untuk pelaksana yang dihentikan.
- Acara untuk menyelesaikan pertanyaan SQL, dan acara pekerjaan, panggung, dan tugas terkait.

Untuk meluncurkan cluster dengan penggulungan dan pemadatan diaktifkan

1. Buat `spark-configuration.json` file dengan konfigurasi berikut.

```
[
   {
     "Classification": "spark-defaults",
        "Properties": {
           "spark.eventLog.rolling.enabled": true,
           "spark.history.fs.eventLog.rolling.maxFilesToRetain": 1
        }
   }
]
```

2. Buat cluster Anda dengan konfigurasi pemadatan bergulir Spark sebagai berikut.

```
aws emr create-cluster \
--release-label emr-6.6.0 \
--instance-type m4.large \
--instance-count 2 \
--use-default-roles \
--configurations file://spark-configuration.json
```

Pertimbangan dan batasan

Akses sekali klik ke antarmuka pengguna aplikasi persisten saat ini memiliki batasan berikut.

- Akan ada setidaknya dua menit penundaan ketika detail aplikasi muncul pada UI Server Riwayat Spark.
- Fitur ini bekerja hanya ketika direktori log peristiwa untuk aplikasi berada dalam HDFS. Secara default, Amazon EMR menyimpan log peristiwa di dalam direktori HDFS. Jika Anda mengubah direktori default ke sistem file yang berbeda, seperti Amazon S3, fitur ini tidak akan bekerja.
- Fitur ini saat ini tidak tersedia untuk klaster EMR dengan beberapa simpul utama atau untuk klaster EMR yang terintegrasi dengan AWS Lake Formation.
- Untuk mengaktifkan akses satu klik ke antarmuka pengguna aplikasi persisten, Anda harus memiliki izin untuk `DescribeCluster` tindakan untuk Amazon EMR. Jika Anda tolak izin utama IAM untuk tindakan ini, dibutuhkan sekitar lima menit bagi perubahan izin tersebut untuk menyebar.
- Jika Anda mengkonfigurasi ulang aplikasi dalam sebuah klaster berjalan, riwayat aplikasi akan tidak tersedia melalui UI aplikasi.

- Untuk masing-masingAkun AWS, batas default untuk UI aplikasi aktif adalah 200.
- Anda dapat mengakses UI aplikasi dari konsol di Wilayah AS Timur (Virginia N.), Wilayah AS Barat (California N.), Wilayah Kanada (Tengah), Uni Eropa (Frankfurt, Irlandia, London, Paris, Stockholm), Asia Pasifik (Mumbai, Seoul, Singapura, Sydney, dan Tokyo), Amerika Selatan (São Paulo), China (Beijing) yang dioperasikan oleh Sinnet, dan China (Ningxia) yang dioperasikan oleh NWW CD.

Melihat riwayat aplikasi tingkat tinggi

> **Note**
>
> Sebaiknya gunakan antarmuka aplikasi persisten untuk meningkatkan pengalaman pengguna yang mempertahankan riwayat aplikasi hingga 30 hari. Riwayat aplikasi tingkat tinggi yang dijelaskan di halaman ini tidak tersedia di konsol EMR Amazon baru (https://console.aws.amazon.com/emr). Untuk informasi selengkapnya, lihat Melihat antarmuka pengguna aplikasi persisten.

Dengan Amazon EMR merilis 5.8.0 hingga 5.36.0 dan 6.x rilis hingga 6.8.0, Anda dapat melihat riwayat aplikasi tingkat tinggi dari tab antarmuka pengguna Aplikasi di konsol EMR Amazon lama. Antarmuka pengguna Aplikasi Amazon EMR menyimpan ringkasan riwayat aplikasi selama 7 hari setelah aplikasi selesai.

Pertimbangan dan batasan

Pertimbangkan batasan berikut saat Anda menggunakan tab Antarmuka pengguna Aplikasi di konsol EMR Amazon lama.

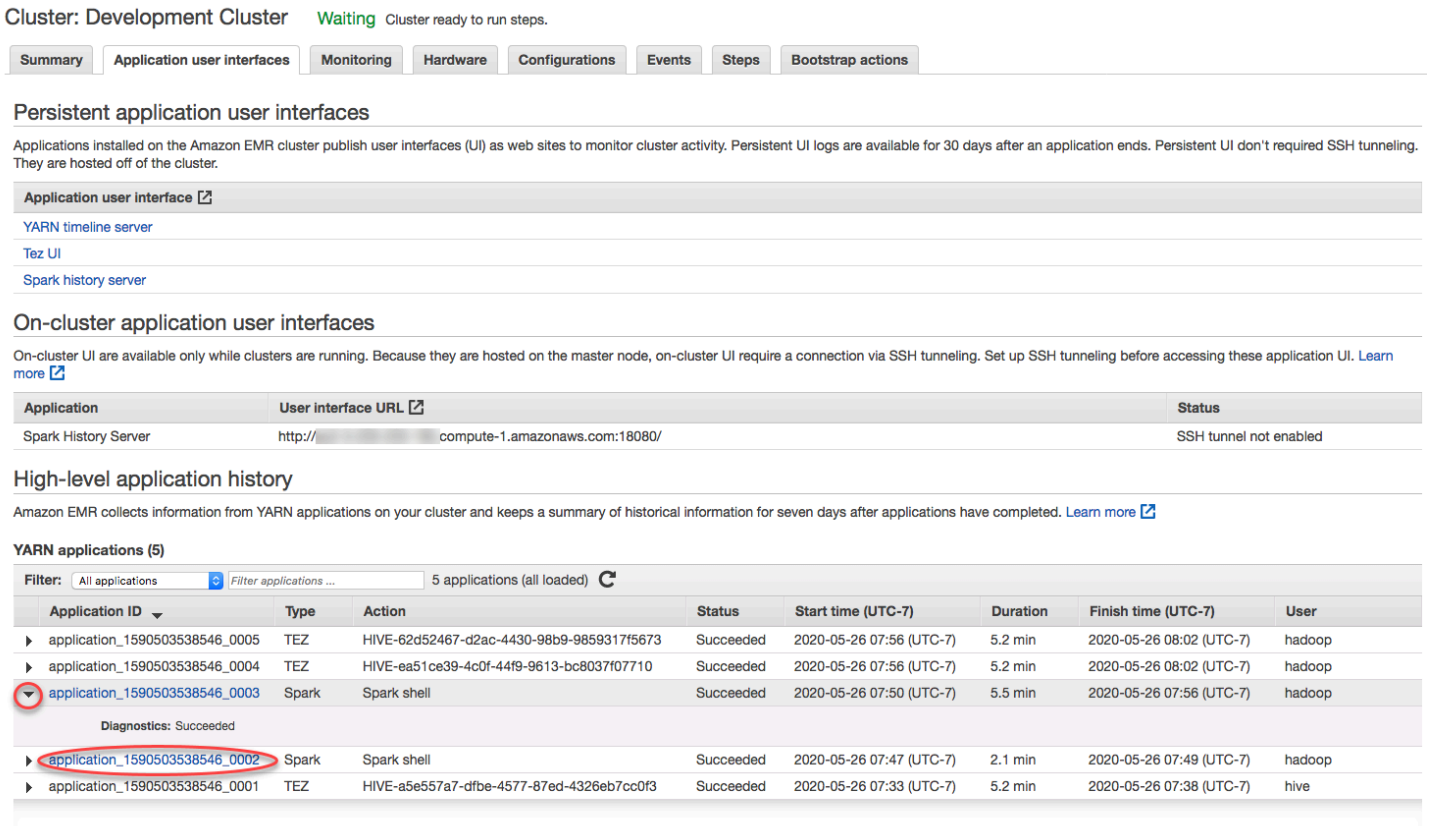
- Anda hanya dapat mengakses fitur riwayat aplikasi tingkat tinggi saat menggunakan Amazon EMR rilis 5.8.0 hingga 5.36.0 dan 6.x rilis hingga 6.8.0. Efektif 23 Januari 2023, Amazon EMR akan menghentikan riwayat aplikasi tingkat tinggi untuk semua versi. Jika Anda menggunakan Amazon EMR versi 5.25.0 atau lebih tinggi, kami sarankan Anda menggunakan antarmuka pengguna aplikasi persisten sebagai gantinya.
- Fitur riwayat aplikasi tingkat tinggi tidak mendukung aplikasi Spark Streaming.
- Akses sekali klik ke antarmuka pengguna aplikasi persisten saat ini tidak tersedia untuk kluster EMR Amazon dengan beberapa node master atau untuk kluster EMR Amazon yang terintegrasi dengannya. AWS Lake Formation

Contoh: Melihat riwayat aplikasi tingkat tinggi

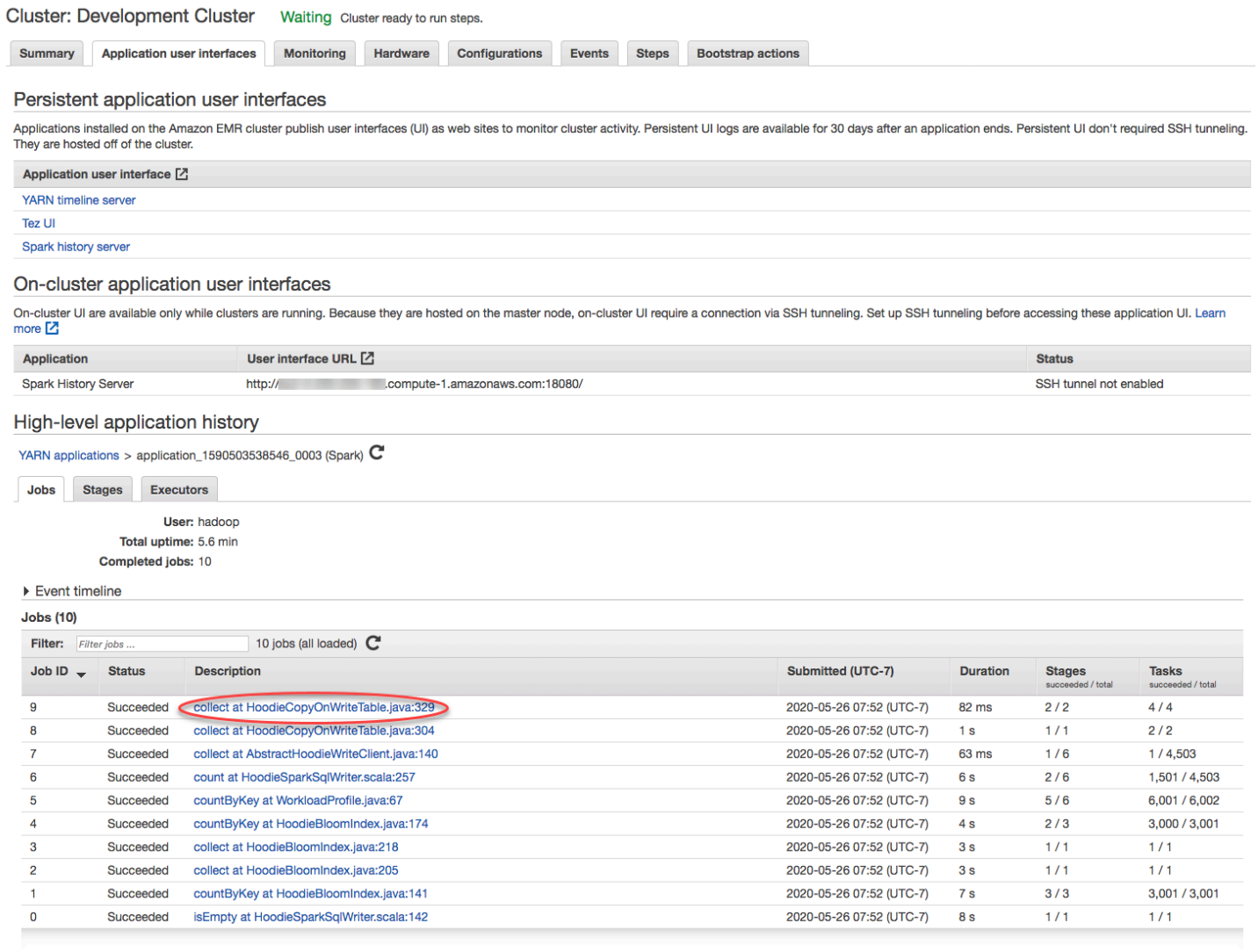
Urutan berikut menunjukkan penelusuran melalui aplikasi Spark atau YARN ke detail pekerjaan menggunakan tab antarmuka pengguna Aplikasi pada halaman detail cluster konsol lama.

Untuk melihat detail klaster, pilih Nama klaster dari daftar Klaster. Untuk melihat informasi tentang log kontainer YARN, Anda harus mengaktifkan logging untuk klaster Anda. Untuk informasi selengkapnya, lihat Mengkonfigurasi logging dan debug klaster. Untuk riwayat aplikasi Spark, informasi yang diberikan dalam tabel ringkasan hanya merupakan bagian dari informasi yang tersedia melalui UI server riwayat Spark.

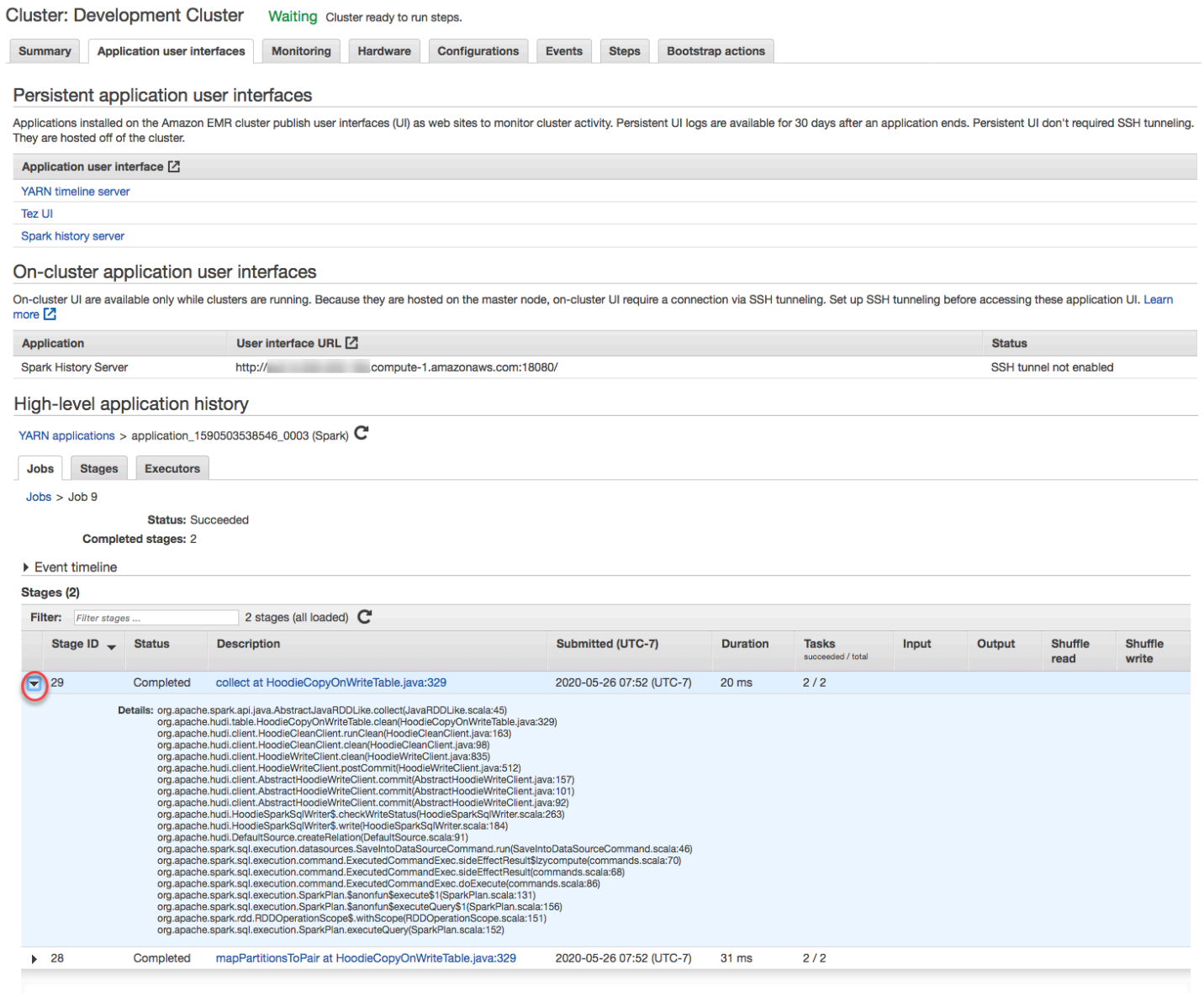
Di tab Antarmuka pengguna aplikasi di bawah Riwayat aplikasi tingkat tinggi, Anda dapat memperluas baris untuk menampilkan ringkasan diagnostik untuk aplikasi Spark atau memilih tautan ID Aplikasi untuk melihat detail tentang aplikasi yang berbeda.



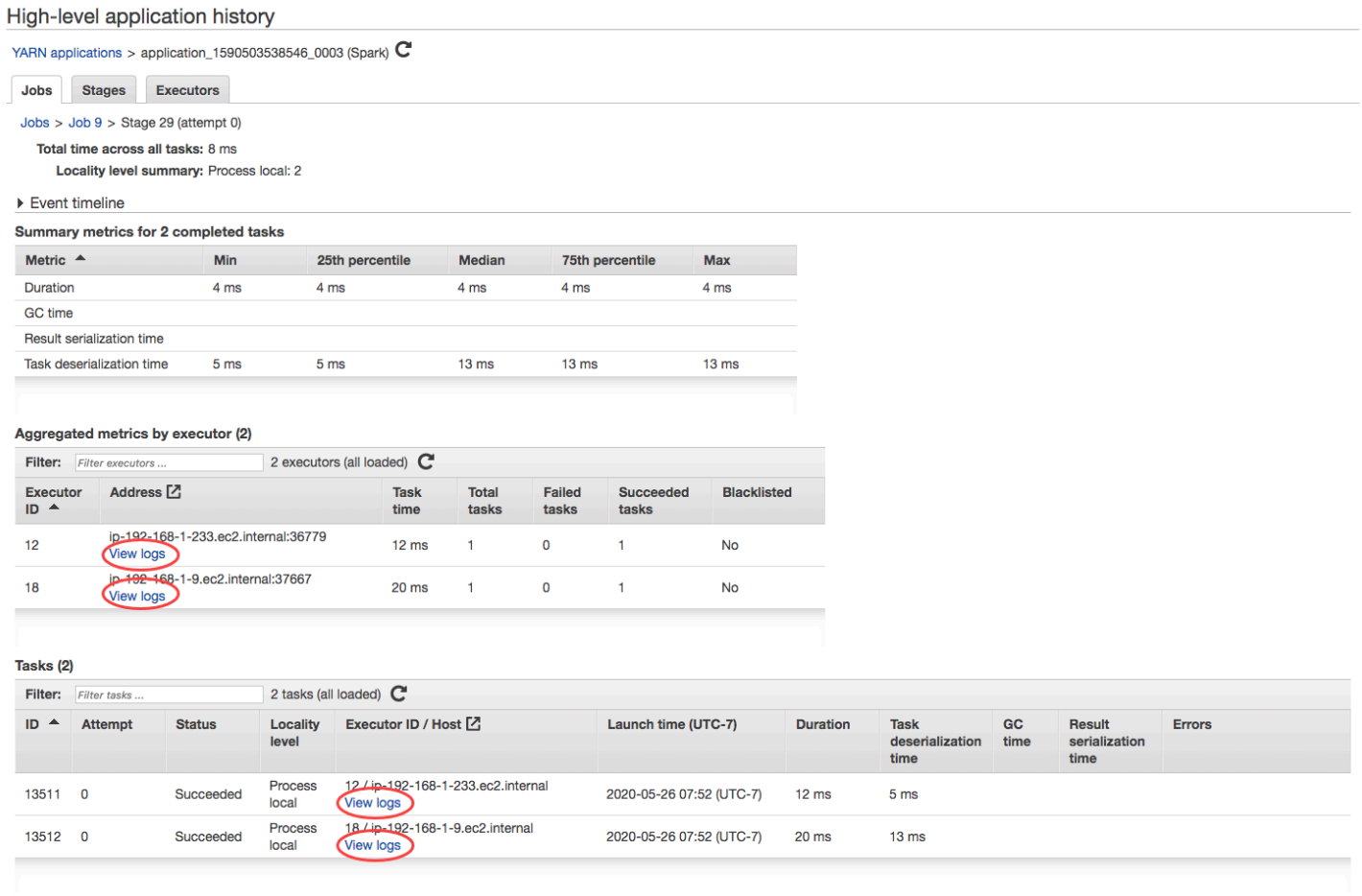
Bila Anda memilih tautan ID Aplikasi, UI berubah untuk menampilkan detail Aplikasi YARN untuk aplikasi tersebut. Di tab Pekerjaan pada detail Aplikasi YARN, Anda dapat memilih tautan Deskripsi agar pekerjaan menampilkan detail untuk pekerjaan tersebut.



Pada halaman detail pekerjaan, Anda dapat memperluas informasi tentang tahapan pekerjaan individu, dan kemudian pilih tautan Deskripsi untuk melihat detail tahapan.



Pada halaman detail tahapan, Anda dapat melihat metrik kunci untuk tugas dan pelaksana tahapan. Anda juga dapat melihat log tugas dan pelaksana menggunakan tautan Lihat log.



Melihat berkas log

Amazon EMR dan Hadoop menghasilkan berkas log yang melaporkan status pada klaster. Secara default, ini ditulis ke simpul utama dalam `/mnt/var/log/` direktori. Tergantung pada cara Anda mengkonfigurasi klaster Anda ketika Anda meluncurkannya, log ini juga dapat diarsipkan ke Amazon S3 dan dapat dilihat melalui alat debugging grafis.

Ada banyak jenis log yang ditulis ke simpul utama. Amazon EMR menulis log langkah, tindakan bootstrap, dan status instans. Apache Hadoop menulis log untuk melaporkan pengolahan pekerjaan, tugas, dan upaya tugas. Hadoop juga mencatat log dari daemon nya. Untuk informasi lebih lanjut tentang log yang ditulis oleh Hadoop, buka http://hadoop.apache.org/docs/stable/ hadoop-project-dist /hadoop-common/ ClusterSetup .html.

Lihat file log pada simpul utama

Tabel berikut mencantumkan beberapa file log yang akan Anda temukan di simpul utama.

Lokasi	Deskripsi
/emr/instance-controller/log/bootstrap-action	Log ditulis selama pemrosesan tindakan bootstrap.
/mnt/var/log/ hadoop-state-pusher	Log ditulis oleh proses pendorong status Hadoop.
/emr/instance-controller/log	Log pengendali instans.
/emr/instance-state	Log status instans. Ini berisi informasi tentang CPU, status memori, dan utas pengumpul sampah dari simpul tersebut.
/emr/layanan-pengasuh	Log ditulis oleh proses pengasuh layanan.
/mnt/var/log/*aplikasi*	Log khusus untuk aplikasi seperti Hadoop, Spark, atau Hive.
/mnt/var/log/hadoop/steps/*N*	Log langkah yang berisi informasi tentang pengolahan langkah. Nilai dari *N* menunjukk an stepId yang ditetapkan oleh Amazon EMR. Sebagai contoh, sebuah klaster memiliki dua langkah: `s-1234ABCDEFGH` dan `s-5678IJK LMNOP` . Langkah pertama terletak di `/mnt/ var/log/hadoop/steps/s-1234ABCD EFGH/` dan langkah kedua di `/mnt/var/ log/hadoop/steps/s-5678IJKL MNOP/` . Log langkah yang ditulis oleh Amazon EMR adalah sebagai berikut. • pengendali — Informasi tentang pengolaha n langkah. Jika langkah Anda gagal saat memuat, Anda dapat menemukan jejak tumpukan dalam log ini.

Lokasi	Deskripsi
	• syslog — Menjelaskan eksekusi pekerjaan Hadoop dalam langkah tersebut. • stderr — Saluran kesalahan standar Hadoop saat memproses langkah. • stdout — Saluran output standar Hadoop saat memproses langkah.

Untuk melihat file log pada node utama dengan fileAWS CLI.

1. Gunakan SSH untuk terhubung ke node utama seperti yang dijelaskan dalamConnect ke node utama menggunakan SSH.
2. Buka direktori yang berisi informasi berkas log yang ingin Anda lihat. Tabel sebelumnya memberikan daftar jenis berkas log yang tersedia dan tempat Anda dapat menemukannya. Contoh berikut menunjukkan perintah untuk membuka log langkah dengan sebuah ID, `s-1234ABCDEFGH`.

   ```
   cd /mnt/var/log/hadoop/steps/s-1234ABCDEFGH/
   ```

3. Gunakan penampil file pilihan Anda untuk melihat berkas log. Contoh berikut menggunakan perintah `less` Linux untuk melihat berkas log `controller`.

   ```
   less controller
   ```

Melihat berkas log yang diarsipkan ke Amazon S3

Secara default, klaster Amazon EMR yang diluncurkan menggunakan konsol secara otomatis mengarsipkan berkas log ke Amazon S3. Anda dapat menentukan jalur log Anda sendiri, atau Anda dapat mengizinkan konsol untuk secara otomatis membuat jalur log untuk Anda. Untuk klaster yang diluncurkan menggunakan CLI atau API, Anda harus mengkonfigurasi log pengarsipan Amazon S3 secara manual.

Ketika Amazon EMR dikonfigurasi untuk mengarsipkan berkas log ke Amazon S3, ia menyimpan file dalam lokasi S3 yang Anda tentukan, di folder /*cluster-id*/, yaitu dimana *cluster-id* merupakan ID klaster.

Tabel berikut mencantumkan beberapa berkas log yang akan Anda temukan pada Amazon S3.

Lokasi	Deskripsi
/*cluster-id* /node/	Log simpul, termasuk tindakan bootstrap, status instans, dan log aplikasi untuk simpul. Log untuk setiap simpul disimpan dalam folder berlabel dengan pengenal instans EC2 dari simpul tersebut.
/*cluster-id* /node/*instance-id* /*application*	Log yang dibuat oleh setiap aplikasi atau daemon terkait dengan suatu aplikasi. Sebagai contoh, log server Hive terletak di *cluster-id* /node/*instance-id* /hive/hive-server.log .
/*cluster-id* /steps/*step-id*/	Log langkah yang berisi informasi tentang pengolahan langkah. Nilai dari *step-id* menunjukkan ID langkah yang ditetapkan oleh Amazon EMR. Sebagai contoh, sebuah klaster memiliki dua langkah: s-1234ABCDEFGH dan s-5678IJKLMNOP . Langkah pertama terletak di /mnt/var/log/hadoop/steps/s-1234ABCDEFGH/ dan langkah kedua di /mnt/var/log/hadoop/steps/s-5678IJKLMNOP/ . Log langkah yang ditulis oleh Amazon EMR adalah sebagai berikut. • pengendali — Informasi tentang pengolahan langkah. Jika langkah Anda gagal saat memuat, Anda dapat menemukan jejak tumpukan dalam log ini. • syslog — Menjelaskan eksekusi pekerjaan Hadoop dalam langkah tersebut.

Lokasi	Deskripsi
	• stderr — Saluran kesalahan standar Hadoop saat memproses langkah. • stdout — Saluran output standar Hadoop saat memproses langkah.
/*cluster-id* /containers	Log kontainer aplikasi. Log untuk setiap aplikasi YARN disimpan di lokasi ini.
/*cluster-id* /hadoop-mapreduce/	Log yang berisi informasi tentang detail konfigurasi dan riwayat pekerjaan MapReduce pekerjaan.

Untuk melihat file log yang diarsipkan ke Amazon S3 dengan konsol Amazon S3

1. Masuk ke AWS Management Console dan buka konsol Amazon S3 di https://console.aws.amazon.com/s3/.
2. Buka bucket S3 yang ditentukan ketika Anda mengkonfigurasi klaster untuk mengarsipkan berkas log di Amazon S3.
3. Buka berkas log yang berisi informasi yang ingin ditampilkan. Tabel sebelumnya memberikan daftar jenis berkas log yang tersedia dan tempat Anda dapat menemukannya.
4. Download objek berkas log untuk melihatnya. Untuk melihat instruksi, lihat Mengunduh objek.

Melihat berkas log dalam alat debugging

Amazon EMR tidak secara otomatis mengaktifkan alat debugging. Anda harus mengkonfigurasi ini ketika Anda meluncurkan klaster. Perhatikan bahwa konsol EMR Amazon baru tidak menawarkan alat debugging.

Untuk melihat log cluster dengan konsol lama

1. Arahkan ke konsol EMR Amazon baru dan pilih Beralih ke konsol lama dari navigasi samping. Untuk informasi selengkapnya tentang apa yang diharapkan saat beralih ke konsol lama, lihat Menggunakan konsol lama.
2. Dari halaman Daftar Klaster, pilih ikon detail di samping klaster yang ingin Anda lihat.

Hal ini memunculkan halaman Detail Klaster. Di bagian Langkah-Langkah, tautan di sebelah kanan setiap langkah menampilkan berbagai jenis log yang tersedia untuk langkah tersebut. Log ini dihasilkan oleh Amazon EMR.

3. Untuk melihat daftar pekerjaan Hadoop yang dikaitkan dengan langkah tertentu, pilih tautan Lihat Pekerjaan di sebelah kanan langkah.
4. Untuk melihat daftar tugas Hadoop yang dikaitkan dengan pekerjaan tertentu, pilih tautan Lihat Tugas di sebelah kanan pekerjaan.
5. Untuk melihat daftar upaya yang telah dijalankan tugas ketika mencoba untuk menyelesaikan, pilih tautan Tampilkan Upaya di sebelah kanan tugas.
6. Untuk melihat log yang dihasilkan oleh upaya tugas, pilih tautan stderr, stdout, dan syslog di sebelah kanan upaya tugas.

Alat debugging menampilkan tautan ke berkas log setelah Amazon EMR mengunggah berkas log ke bucket Anda di Amazon S3. Karena berkas log diunggah ke Amazon S3 setiap 5 menit, dapat memakan waktu beberapa menit sampai unggahan berkas log selesai setelah langkah tersebut selesai.

Amazon EMR secara berkala memperbarui status pekerjaan, tugas, dan upaya tugas Hadoop dalam alat debugging. Anda dapat mengklik Refresh List di panel debugging untuk mendapatkan up-to-date status terbanyak dari item ini.

Melihat instans klaster di Amazon EC2

Untuk membantu Anda mengelola sumber daya Anda, Amazon EC2 memungkinkan Anda untuk menetapkan metadata ke sumber daya dalam bentuk tanda. Setiap tanda Amazon EC2 terdiri dari kunci dan nilai. Tanda mengizinkan Anda untuk mengategorikan sumber daya Amazon EC2 Anda dengan berbagai cara, misalnya, berdasarkan tujuan, pemilik, atau lingkungan.

Anda dapat mencari dan mem-filter sumber daya berdasarkan tanda. Tag yang Anda tetapkan ke sumber daya melalui AWS akun Anda hanya tersedia untuk Anda. Akun lain yang berbagi sumber daya yang sama tidak dapat melihat tag Anda.

Amazon EMR secara otomatis menandai setiap instans EC2 yang diluncurkan dengan pasangan nilai kunci. Kunci mengidentifikasi cluster dan grup instance yang menjadi milik instance. Ini memudahkan untuk memfilter instans EC2 Anda untuk ditampilkan, misalnya, hanya instance yang termasuk dalam cluster tertentu, atau untuk menampilkan semua instance yang sedang berjalan di grup instans untuk

tugas tersebut. Ini sangat berguna jika Anda menjalankan beberapa cluster secara bersamaan atau mengelola sejumlah besar instans EC2.

Berikut ini adalah pasangan nilai kunci yang telah ditetapkan Amazon EMR:

Kunci	Nilai	Definisi nilai
aws:elasticmapreduce:job-flow-id	*job-flow-identifier*	ID cluster tempat instance disediakan untuk. Itu muncul dalam format `j-XXXXXXXXXXXXX` dan bisa mencapai 256 karakter.
aws:elasticmapreduce:instance-group-role	*group-role*	Jenis grup contoh, dimasukkan sebagai salah satu nilai berikut:`master`,`core`, atau`task`.

Anda dapat melihat dan mem-filter pada tanda yang ditambahkan oleh Amazon EMR. Untuk informasi selengkapnya, lihat Menggunakan tanda dalam Panduan Pengguna Amazon EC2 untuk Instans Linux. Karena tanda yang ditetapkan oleh Amazon EMR adalah tanda sistem dan tidak dapat diedit atau dihapus, bagian pada menampilkan dan mem-filter tag adalah yang paling relevan.

> **Note**
>
> Amazon EMR menambahkan tag ke instans EC2 saat statusnya diperbarui ke Running. Jika latensi terjadi antara waktu instans EC2 disediakan dan waktu statusnya disetel ke Running, tag yang disetel Amazon EMR akan muncul setelah instance dimulai. Jika Anda tidak melihat tanda, tunggu beberapa menit dan segarkan tampilan.

CloudWatch peristiwa dan metrik

Gunakan peristiwa dan metrik untuk melacak aktivitas dan kesehatan klaster Amazon EMR. Peristiwa berguna untuk memantau kejadian tertentu dalam sebuah klaster - misalnya, ketika sebuah klaster berubah status dari mulai menjadi sedang berjalan. Metrik berguna untuk memantau nilai tertentu - misalnya, persentase ruang disk yang tersedia yang digunakan HDFS dalam sebuah cluster.

Untuk informasi selengkapnya tentang CloudWatch Acara, lihat Panduan Pengguna CloudWatch Acara Amazon. Untuk informasi selengkapnya tentang CloudWatch metrik, lihat Menggunakan

CloudWatch metrik Amazon dan Membuat CloudWatch alarm Amazon di Panduan Pengguna Amazon CloudWatch .

Topik

- Memantau metrik Amazon EMR dengan CloudWatch
- Memantau peristiwa EMR Amazon dengan CloudWatch
- Menanggapi peristiwa CloudWatch

Memantau metrik Amazon EMR dengan CloudWatch

Metrik diperbarui setiap lima menit dan secara otomatis dikumpulkan dan didorong ke setiap CloudWatch klaster EMR Amazon. Interval ini tidak dapat dikonfigurasi. Tidak ada biaya untuk metrik EMR Amazon yang dilaporkan. CloudWatch Metrik titik data lima menit ini diarsipkan selama 63 hari, dan setelahnya data tersebut dibuang.

Bagaimana cara menggunakan metrik Amazon EMR?

Tabel berikut menunjukkan penggunaan umum untuk metrik yang dilaporkan oleh Amazon EMR. Berikut ini adalah saran agar Anda dapat mulai, bukan daftar komprehensif. Untuk daftar lengkap metrik yang dilaporkan oleh Amazon EMR, lihat Metrik dilaporkan oleh Amazon EMR di CloudWatch.

Bagaimana cara saya?	Metrik Terkait
Melacak kemajuan klaster saya	Melihat metrik `RunningMapTasks` , `RemainingMapTasks` , `RunningReduceTasks` , dan `RemainingReduceTasks` .
Mendeteksi klaster yang menganggur	Metrik `IsIdle` melacak apakah klaster sedang siaga, namun bukan merupakan tugas yang sedang berjalan. Anda dapat mengatur alarm untuk berbunyi ketika klaster telah menganggur selama jangka waktu tertentu, seperti tiga puluh menit.
Mendeteksi ketika sebuah simpul kehabisan penyimpanan	`MRUnhealthyNodes` Metrik melacak ketika satu atau lebih node inti atau tugas kehabisan

Bagaimana cara saya?	Metrik Terkait
	penyimpanan disk lokal dan transisi ke status `UNHEALTHY` YARN. Misalnya, node inti atau tugas kehabisan ruang disk dan tidak akan dapat menjalankan tugas.
Mendeteksi ketika cluster kehabisan penyimpanan	`HDFSUtilization` Metrik memantau kapasitas HDFS gabungan cluster, dan dapat memerlukan pengubahan ukuran cluster untuk menambahkan lebih banyak node inti. Misalnya, pemanfaatan HDFS tinggi, yang dapat mempengaruhi pekerjaan dan kesehatan cluster.
Mendeteksi saat cluster berjalan pada kapasitas berkurang	`MRLostNodes` Metrik melacak ketika satu atau lebih inti atau node tugas tidak dapat berkomunikasi dengan node master. Misalnya, inti atau node tugas tidak dapat dijangkau oleh node master.

Untuk informasi selengkapnya, lihat Klaster berakhir dengan NO_SLAVE_LEFT dan simpul inti FAILED_BY_MASTER dan AWSSupport-AnalyzeEmrLogs.

Akses CloudWatch metrik untuk Amazon EMR

Anda dapat melihat metrik yang dilaporkan Amazon EMR menggunakan CloudWatch konsol Amazon EMR atau konsol. CloudWatch Anda juga dapat mengambil metrik menggunakan perintah CloudWatch `mon-get-stats` CLI atau API. CloudWatch `GetMetricStatistics` Untuk informasi selengkapnya tentang melihat atau mengambil metrik untuk Amazon EMR menggunakan, CloudWatch lihat Panduan Pengguna Amazon. CloudWatch

> **Note**
>
> Kami telah mendesain ulang konsol EMR Amazon agar lebih mudah digunakan. Lihat Apa yang baru dengan konsol? untuk mempelajari tentang perbedaan antara pengalaman konsol lama dan baru.

New console

Untuk melihat metrik dengan konsol baru

1. Masuk keAWS Management Console, dan buka konsol EMR Amazon di https://console.aws.amazon.com/emr.
2. Di bawah EMR pada EC2 di panel navigasi kiri, pilih Cluster, lalu pilih cluster yang ingin Anda lihat metriknya. Ini membuka halaman detail cluster.
3. Pilih tab Monitoring pada halaman detail cluster. Pilih salah satu status Cluster, status Node, atau opsi Input dan output untuk memuat laporan tentang kemajuan dan kesehatan cluster.
4. Setelah Anda memilih metrik untuk dilihat, Anda dapat memperbesar setiap grafik. Untuk memfilter kerangka waktu grafik Anda, pilih opsi yang telah diisi sebelumnya atau pilih Kustom.

Old console

Untuk melihat metrik dengan konsol lama

1. Buka konsol Amazon EMR di https://console.aws.amazon.com/elasticmapreduce/.
2. Untuk melihat metrik untuk sebuah klaster, pilih klaster untuk menampilkan panel Ringkasan.
3. Pilih Pemantauan untuk melihat informasi tentang klaster tersebut. Pilih salah satu tab bernama Status Klaster, Pemetaan/Peredaman, Status Simpul, atau IO untuk memuat laporan tentang kemajuan dan kesehatan klaster.
4. Setelah memilih metrik untuk dilihat, Anda dapat memilih ukuran grafik. Edit Mulai dan Akhir untuk mem-filter metrik ke kerangka waktu tertentu.

Metrik dilaporkan oleh Amazon EMR di CloudWatch

Tabel berikut mencantumkan metrik yang dilaporkan Amazon EMR di konsol dan mendorong ke. CloudWatch

Metrik Amazon EMR

Amazon EMR mengirimkan data untuk beberapa metrik ke. CloudWatch Semua klaster Amazon EMR secara otomatis mengirim metrik dalam interval lima menit. Metrik diarsipkan selama dua minggu; setelah periode itu, data akan dibuang.

Namespace `AWS/ElasticMapReduce` mencakup metrik berikut.

> **Note**
>
> Amazon EMR menarik metrik dari klaster. Jika klaster menjadi tidak terjangkau, tidak ada metrik yang dilaporkan sampai klaster tersebut tersedia kembali.

Metrik berikut tersedia untuk klaster yang menjalankan versi Hadoop 2.x.

Metrik	Deskripsi
Status Cluster	
IsIdle	Menunjukkan bahwa klaster tidak lagi melakukan pekerjaan , tetapi masih hidup dan menimbulkan biaya. Diatur ke 1 jika tidak ada tugas yang berjalan dan tidak ada pekerjaan yang berjalan, dan diatur ke 0 jika sebaliknya. Nilai ini diperiksa pada interval lima menit dan nilai 1 hanya menunjukkan bahwa klaster tersebut menganggur ketika diperiksa, bukan bahwa klaster tersebut menganggur selama lima menit tersebut. Untuk menghindari positif yang salah, Anda harus menyalakan alarm ketika nilai ini 1 selama lebih dari satu pemeriksaan 5 menit berturut-turut. Misalnya, Anda mungkin menyalakan alarm pada nilai ini jika telah 1 selama tiga puluh menit atau lebih. Kasus penggunaan: Memantau performa klaster Unit: Boolean
ContainerAllocated	Jumlah wadah sumber daya yang dialokasikan oleh. ResourceManager Kasus penggunaan: Memantau kemajuan klaster Unit: Jumlah
ContainerReserved	Jumlah kontainer yang disimpan. Kasus penggunaan: Memantau kemajuan klaster

Metrik	Deskripsi
	Unit: Jumlah
ContainerPending	Jumlah kontainer dalam antrean yang belum dialokasikan. Kasus penggunaan: Memantau kemajuan klaster Unit: Jumlah
ContainerPendingRatio	Rasio kontainer yang tertunda dengan kontainer yang dialokasikan (ContainerPendingRatio = ContainerPending / ContainerAllocated). Jika ContainerAllocated = 0, maka ContainerPendingRatio =ContainerPending. Nilai Container PendingRatio mewakili angka, bukan persentase. Nilai ini berguna untuk menskalakan sumber daya klaster berdasark an perilaku alokasi kontainer. Unit: Jumlah
AppsCompleted	Jumlah aplikasi yang dikirimkan ke YARN yang telah selesai. Kasus penggunaan: Memantau kemajuan klaster Unit: Jumlah
AppsFailed	Jumlah aplikasi yang dikirimkan ke YARN yang gagal diselesaikan. Kasus penggunaan: Memantau kemajuan klaster, Memantau kesehatan klaster Unit: Jumlah

Metrik	Deskripsi
AppsKilled	Jumlah aplikasi yang dikirimkan ke YARN yang telah dimatikan. Kasus penggunaan: Memantau kemajuan klaster, Memantau kesehatan klaster Unit: Jumlah
AppsPending	Jumlah aplikasi yang dikirimkan ke YARN yang berada dalam status tertunda. Kasus penggunaan: Memantau kemajuan klaster Unit: Jumlah
AppsRunning	Jumlah aplikasi yang dikirimkan ke YARN yang sedang berjalan. Kasus penggunaan: Memantau kemajuan klaster Unit: Jumlah
AppsSubmitted	Jumlah aplikasi yang dikirimkan ke YARN. Kasus penggunaan: Memantau kemajuan klaster Unit: Jumlah
Status Node	
CoreNodesRunning	Jumlah simpul inti yang bekerja. Titik data untuk metrik ini hanya dilaporkan apabila grup instans yang sesuai tersedia. Kasus penggunaan: Memantau kesehatan klaster Unit: Jumlah

Metrik	Deskripsi
CoreNodesPending	Jumlah simpul inti yang menunggu untuk ditugaskan. Semua simpul inti yang diminta mungkin tidak segera tersedia; metrik ini melaporkan permintaan yang tertunda. Titik data untuk metrik ini hanya dilaporkan apabila grup instans yang sesuai tersedia. Kasus penggunaan: Memantau kesehatan klaster Unit: Jumlah
LiveDataNodes	Persentase simpul data yang menerima pekerjaan dari Hadoop. Kasus penggunaan: Memantau kesehatan klaster Unit: Persen
MR TotalNodes	Jumlah node yang saat ini tersedia untuk MapReduce pekerjaan. Setara dengan metrik YARN `mapred.resourcemanager.TotalNodes` . Kasus penggunaan: Memantau kemajuan klaster Unit: Jumlah
MR ActiveNodes	Jumlah node yang saat ini menjalankan MapReduce tugas atau pekerjaan. Setara dengan metrik YARN `mapred.resourcemanager.NoOfActiveNodes` . Kasus penggunaan: Memantau kemajuan klaster Unit: Jumlah

Metrik	Deskripsi
MR LostNodes	Jumlah node yang dialokasikan untuk MapReduce yang telah ditandai dalam keadaan LOST. Setara dengan metrik YARN `mapred.resourcemanager.NoOfLostNodes` . Kasus penggunaan: Memantau kesehatan klaster, Memantau kemajuan klaster Unit: Jumlah
MR UnhealthyNodes	Jumlah node yang tersedia untuk MapReduce pekerjaan yang ditandai dalam keadaan TIDAK SEHAT. Setara dengan metrik YARN `mapred.resourceman ager.NoOfUnhealthyNodes` . Kasus penggunaan: Memantau kemajuan klaster Unit: Jumlah
MR DecommissionedNodes	Jumlah node yang dialokasikan untuk MapReduce aplikasi yang telah ditandai dalam keadaan DECOMMISSIONED. Setara dengan metrik YARN `mapred.resourceman ager.NoOfDecommissionedNodes` . Kasus penggunaan: Memantau kesehatan klaster, Memantau kemajuan klaster Unit: Jumlah
MR RebootedNodes	Jumlah node yang tersedia untuk MapReduce yang telah di-boot ulang dan ditandai dalam status REBOOTED. Setara dengan metrik YARN `mapred.resourceman ager.NoOfRebootedNodes` . Kasus penggunaan: Memantau kesehatan klaster, Memantau kemajuan klaster Unit: Jumlah

Metrik	Deskripsi
MultiMasterInstanceGroupNodesRunning	Jumlah simpul utama yang sedang berjalan. Kasus penggunaan: Memantau kegagalan dan penggantian simpul utama Unit: Jumlah
MultiMasterInstanceGroupNodesRunningPercentage	Persentase simpul utama yang berjalan dibandingkan jumlah instans simpul utama yang diminta. Kasus penggunaan: Memantau kegagalan dan penggantian simpul utama Unit: Persen
MultiMasterInstanceGroupNodesRequested	Jumlah simpul utama yang diminta. Kasus penggunaan: Memantau kegagalan dan penggantian simpul utama Unit: Jumlah
IO	
S3 BytesWritten	Jumlah byte yang ditulis ke Amazon S3. Metrik ini hanya mengumpulkan MapReduce pekerjaan, dan tidak berlaku untuk beban kerja lain di Amazon EMR. Kasus penggunaan: Menganalisis performa klaster, Memantau kemajuan klaster Unit: Jumlah

Metrik	Deskripsi
S3 BytesRead	Jumlah byte yang dibaca dari Amazon S3. Metrik ini hanya mengumpulkan MapReduce pekerjaan, dan tidak berlaku untuk beban kerja lain di Amazon EMR. Kasus penggunaan: Menganalisis performa klaster, Memantau kemajuan klaster Unit: Jumlah
HDFSUtilization	Persentase penyimpanan HDFS yang saat ini digunakan. Kasus penggunaan: Menganalisis performa klaster Unit: Persen
HDFS BytesRead	Jumlah byte yang dibaca dari HDFS. Metrik ini hanya mengumpulkan MapReduce pekerjaan, dan tidak berlaku untuk beban kerja lain di Amazon EMR. Kasus penggunaan: Menganalisis performa klaster, Memantau kemajuan klaster Unit: Jumlah
HDFS BytesWritten	Jumlah byte yang ditulis ke HDFS. Metrik ini hanya mengumpulkan MapReduce pekerjaan, dan tidak berlaku untuk beban kerja lain di Amazon EMR. Kasus penggunaan: Menganalisis performa klaster, Memantau kemajuan klaster Unit: Jumlah
MissingBlocks	Jumlah blok yang tidak ada replika HDFS. Ini mungkin blok rusak. Kasus penggunaan: Memantau kesehatan klaster Unit: Jumlah

Metrik	Deskripsi
CorruptBlocks	Jumlah blok yang HDFS laporkan sebagai rusak. Kasus penggunaan: Memantau kesehatan klaster Unit: Jumlah
TotalLoad	Jumlah total transfer data secara bersamaan. Kasus penggunaan: Memantau kesehatan klaster Unit: Jumlah
MemoryTotalMB	Total jumlah memori dalam klaster. Kasus penggunaan: Memantau kemajuan klaster Unit: Jumlah
MemoryReservedMB	Jumlah memori yang direservasi. Kasus penggunaan: Memantau kemajuan klaster Unit: Jumlah
MemoryAvailableMB	Jumlah memori yang tersedia untuk dialokasikan. Kasus penggunaan: Memantau kemajuan klaster Unit: Jumlah
BENANG MemoryAvailablePercentage	Persentase sisa memori yang tersedia untuk YARN (YARN MemoryAvailablePercentage = MemoryAvailable MB/ MemoryTotalMB). Nilai ini berguna untuk menskalakan sumber daya klaster berdasarkan penggunaan memori YARN. Unit: Persen

Metrik	Deskripsi
MemoryAllocatedMB	Jumlah memori yang dialokasikan ke klaster. Kasus penggunaan: Memantau kemajuan klaster Unit: Jumlah
PendingDeletionBlocks	Jumlah blok yang ditandai untuk dihapus. Kasus penggunaan: Memantau kemajuan klaster, Memantau kesehatan klaster Unit: Jumlah
UnderReplicatedBlocks	Jumlah blok yang perlu direplikasi satu kali atau lebih. Kasus penggunaan: Memantau kemajuan klaster, Memantau kesehatan klaster Unit: Jumlah
DfsPendingReplicationBlocks	Status replikasi blok: blok direplikasi, umur permintaan replikasi, dan permintaan replikasi yang tidak berhasil. Kasus penggunaan: Memantau kemajuan klaster, Memantau kesehatan klaster Unit: Jumlah
CapacityRemainingGB	Jumlah sisa kapasitas disk HDFS. Kasus penggunaan: Memantau kemajuan klaster, Memantau kesehatan klaster Unit: Jumlah

Berikut ini adalah metrik Hadoop 1:

Metrik	Deskripsi
Status Cluster	
IsIdle	Menunjukkan bahwa klaster tidak lagi melakukan pekerjaan , tetapi masih hidup dan menimbulkan biaya. Diatur ke 1 jika tidak ada tugas yang berjalan dan tidak ada pekerjaan yang berjalan, dan diatur ke 0 jika sebaliknya. Nilai ini diperiksa pada interval lima menit dan nilai 1 hanya menunjukkan bahwa klaster tersebut menganggur ketika diperiksa, bukan bahwa klaster tersebut menganggur selama lima menit tersebut. Untuk menghindari positif yang salah, Anda harus menyalakan alarm ketika nilai ini 1 selama lebih dari satu pemeriksaan 5 menit berturut-turut. Misalnya, Anda mungkin menyalakan alarm pada nilai ini jika telah 1 selama tiga puluh menit atau lebih. Kasus penggunaan: Memantau performa klaster Unit: Boolean
JobsRunning	Jumlah pekerjaan di klaster yang sedang berjalan. Kasus penggunaan: Memantau kesehatan klaster Unit: Jumlah
JobsFailed	Jumlah pekerjaan di klaster yang telah gagal. Kasus penggunaan: Memantau kesehatan klaster Unit: Jumlah
Peta/Kurangi	
MapTasksRunning	Jumlah tugas pemetaan yang berjalan untuk setiap pekerjaan. Jika Anda memiliki penjadwal terpasang dan beberapa pekerjaan yang sedang berjalan, beberapa grafik akan dihasilkan.

Metrik	Deskripsi
	Kasus penggunaan: Memantau kemajuan klaster Unit: Jumlah
MapTasksRemaining	Jumlah sisa tugas pemetaan untuk setiap pekerjaan. Jika Anda memiliki penjadwal terpasang dan beberapa pekerjaan yang sedang berjalan, beberapa grafik akan dihasilkan. Tugas pemetaan yang tersisa adalah tugas yang tidak berada dalam salah satu status berikut: Berjalan, Dimatikan, atau Selesai. Kasus penggunaan: Memantau kemajuan klaster Unit: Jumlah
MapSlotsOpen	Kapasitas tugas pemetaan yang tidak terpakai. Ini dihitung sebagai jumlah maksimum tugas pemetaan untuk klaster tertentu, dikurangi jumlah total tugas pemetaan yang saat ini berjalan di klaster tersebut. Kasus penggunaan: Menganalisis performa klaster Unit: Jumlah
RemainingMapTasksPerSlot	Rasio total tugas pemetaan yang tersisa untuk total slot peta yang tersedia di klaster. Kasus penggunaan: Menganalisis performa klaster Unit: Rasio
ReduceTasksRunning	Jumlah tugas peredaman yang berjalan untuk setiap pekerjaan. Jika Anda memiliki penjadwal terpasang dan beberapa pekerjaan yang sedang berjalan, beberapa grafik akan dihasilkan. Kasus penggunaan: Memantau kemajuan klaster Unit: Jumlah

Metrik	Deskripsi
ReduceTasksRemaining	Jumlah tugas peredaman yang tersisa untuk setiap pekerjaan. Jika Anda memiliki penjadwal terpasang dan beberapa pekerjaan yang sedang berjalan, beberapa grafik akan dihasilkan. Kasus penggunaan: Memantau kemajuan klaster Unit: Jumlah
ReduceSlotsOpen	Kapasitas tugas peredaman yang tidak terpakai. Ini dihitung sebagai kapasitas tugas peredaman maksimal untuk klaster tertentu, dikurangi jumlah tugas peredaman yang saat ini berjalan di klaster tersebut. Kasus penggunaan: Menganalisis performa klaster Unit: Jumlah
Status Node	
CoreNodesRunning	Jumlah simpul inti yang bekerja. Titik data untuk metrik ini hanya dilaporkan apabila grup instans yang sesuai tersedia. Kasus penggunaan: Memantau kesehatan klaster Unit: Jumlah
CoreNodesPending	Jumlah simpul inti yang menunggu untuk ditugaskan. Semua simpul inti yang diminta mungkin tidak segera tersedia; metrik ini melaporkan permintaan yang tertunda. Titik data untuk metrik ini hanya dilaporkan apabila grup instans yang sesuai tersedia. Kasus penggunaan: Memantau kesehatan klaster Unit: Jumlah

Metrik	Deskripsi
LiveDataNodes	Persentase simpul data yang menerima pekerjaan dari Hadoop. Kasus penggunaan: Memantau kesehatan klaster Unit: Persen
TaskNodesRunning	Jumlah simpul tugas yang bekerja. Titik data untuk metrik ini hanya dilaporkan apabila grup instans yang sesuai tersedia. Kasus penggunaan: Memantau kesehatan klaster Unit: Jumlah
TaskNodesPending	Jumlah simpul tugas yang menunggu untuk ditugaskan. Semua simpul tugas yang diminta mungkin tidak segera tersedia; metrik ini melaporkan permintaan yang tertunda. Titik data untuk metrik ini hanya dilaporkan apabila grup instans yang sesuai tersedia. Kasus penggunaan: Memantau kesehatan klaster Unit: Jumlah
LiveTaskTrackers	Persentase pelacak tugas yang fungsional. Kasus penggunaan: Memantau kesehatan klaster Unit: Persen
IO	

Metrik	Deskripsi
S3 BytesWritten	Jumlah byte yang ditulis ke Amazon S3. Metrik ini hanya mengumpulkan MapReduce pekerjaan, dan tidak berlaku untuk beban kerja lain di Amazon EMR. Kasus penggunaan: Menganalisis performa klaster, Memantau kemajuan klaster Unit: Jumlah
S3 BytesRead	Jumlah byte yang dibaca dari Amazon S3. Metrik ini hanya mengumpulkan MapReduce pekerjaan, dan tidak berlaku untuk beban kerja lain di Amazon EMR. Kasus penggunaan: Menganalisis performa klaster, Memantau kemajuan klaster Unit: Jumlah
HDFSUtilization	Persentase penyimpanan HDFS yang saat ini digunakan. Kasus penggunaan: Menganalisis performa klaster Unit: Persen
HDFS BytesRead	Jumlah byte yang dibaca dari HDFS. Kasus penggunaan: Menganalisis performa klaster, Memantau kemajuan klaster Unit: Jumlah
HDFS BytesWritten	Jumlah byte yang ditulis ke HDFS. Kasus penggunaan: Menganalisis performa klaster, Memantau kemajuan klaster Unit: Jumlah

Metrik	Deskripsi
MissingBlocks	Jumlah blok yang tidak ada replika HDFS. Ini mungkin blok rusak. Kasus penggunaan: Memantau kesehatan klaster Unit: Jumlah
TotalLoad	Saat ini, jumlah total pembaca dan penulis yang dilaporkan oleh semua DataNodes dalam satu cluster. Kasus penggunaan: Mendiagnosis sejauh mana I/O tinggi mungkin berkontribusi terhadap performa eksekusi pekerjaan yang buruk. Node pekerja yang menjalank an DataNode daemon juga harus melakukan peta dan mengurangi tugas. TotalLoad Nilai tinggi yang terus-men erus dari waktu ke waktu dapat menunjukkan bahwa I/ O yang tinggi mungkin menjadi faktor yang berkontribusi terhadap kinerja yang buruk. Lonjakan sesekali dalam nilai ini biasa terjadi dan biasanya tidak menunjukkan adanya masalah. Unit: Jumlah

Metrik kapasitas klaster

Metrik berikut menunjukkan kapasitas saat ini atau kapasitas target suatu klaster. Metrik ini hanya tersedia saat penskalaan terkelola atau penghentian otomatis diaktifkan.

Untuk klaster yang terdiri dari armada instans, metrik kapasitas klaster diukur dalam `Units`. Untuk klaster yang terdiri dari grup instans, metrik kapasitas klaster diukur dalam `Nodes` atau `VCPU` berdasarkan jenis unit yang digunakan dalam kebijakan penskalaan terkelola. Untuk informasi selengkapnya, lihat Menggunakan penskalaan terkelola EMR dalam Panduan Pengelolaan Amazon EMR.

Metrik	Deskripsi
• TotalUnitsRequested • TotalNodesRequested • TotalVCPURequested	Target jumlah total unit/simpul/vCPU dalam sebuah klaster yang ditentukan oleh penskalaan terkelola. Unit: Jumlah
• TotalUnitsRunning • TotalNodesRunning • TotalVCPURunning	Jumlah total unit/simpul/vCPU saat ini yang tersedia dalam klaster yang sedang berjalan. Ketika ada permintaan perubahan ukuran klaster, metrik ini akan diperbarui setelah instans baru ditambahkan atau dihapus dari klaster. Unit: Jumlah
• CoreUnitsRequested • CoreNodesRequested • CoreVCPURequested	Target jumlah unit/simpul/vCPU INTI dalam sebuah klaster yang ditentukan oleh penskalaan terkelola. Unit: Jumlah
• CoreUnitsRunning • CoreNodesRunning • CoreVCPURunning	Jumlah unit/simpul/vCPU INTI saat ini yang berjalan dalam suatu klaster. Unit: Jumlah
• TaskUnitsRequested • TaskNodesRequested • TaskVCPURequested	Jumlah target unit/simpul/vCPU TUGAS dalam sebuah klaster yang ditentukan oleh penskalaan terkelola. Unit: Jumlah

Metrik	Deskripsi
• `TaskUnitsRunning` • `TaskNodesRunning` • `TaskVCPURunning`	Jumlah unit/simpul/vCPU TUGAS saat ini yang berjalan dalam suatu klaster. Unit: Jumlah

Amazon EMR memancarkan metrik berikut dengan perincian satu menit saat Anda mengaktifkan penghentian otomatis menggunakan kebijakan penghentian otomatis. Beberapa metrik hanya tersedia untuk Amazon EMR versi 6.4.0 dan yang lebih baru. Untuk mempelajari lebih lanjut tentang penghentian otomatis, lihatMenggunakan kebijakan penghentian otomatis.

Metrik	Deskripsi
`TotalNotebookKernels`	Jumlah total kernel notebook yang berjalan dan idle di cluster. Metrik ini hanya tersedia untuk Amazon EMR versi 6.4.0 dan yang lebih baru.
`AutoTerminationIsClusterIdle`	Menunjukkan apakah cluster sedang digunakan . Nilai 0 menunjukkan bahwa cluster digunakan secara aktif oleh salah satu komponen berikut: • Aplikasi YARN • HDFS • Sebuah buku catatan • UI on-cluster, seperti Spark History Server

Metrik	Deskripsi
	Nilai 1 menunjukkan bahwa cluster menganggur. Amazon EMR memeriksa kemalasan cluster berkelanjutan (`AutoTerminationIsClusterIdle` = 1). Jika waktu idle klaster sama dengan `IdleTimeout` nilai dalam kebijakan penghentian otomatis, Amazon EMR akan menghentikan klaster.

Dimensi untuk metrik Amazon EMR

Data Amazon EMR dapat difilter menggunakan salah satu dimensi dalam tabel berikut.

Dimensi	Deskripsi
JobFlowId	Sama seperti ID klaster, yang merupakan pengidentifikasi unik klaster dalam bentuk `j-XXXXXXXXXXXXX` . Temukan nilai ini dengan mengklik klaster yang dimaksud dalam konsol Amazon EMR.

Memantau peristiwa EMR Amazon dengan CloudWatch

Amazon EMR melacak peristiwa dan menyimpan informasi tentangnya hingga tujuh hari di konsol EMR Amazon. Amazon EMR merekam peristiwa ketika ada perubahan dalam status klaster, grup instans, armada instance, kebijakan penskalaan otomatis, atau langkah. Peristiwa menangkap tanggal dan waktu peristiwa terjadi, detail tentang elemen yang terpengaruh, dan titik data penting lainnya.

Tabel berikut mencantumkan peristiwa EMR Amazon, bersama dengan perubahan status atau status yang ditunjukkan peristiwa, tingkat keparahan peristiwa, jenis peristiwa, kode peristiwa, dan pesan peristiwa. Amazon EMR mewakili peristiwa sebagai objek JSON dan secara otomatis mengirimkannya ke aliran acara. Objek JSON penting ketika Anda mengatur aturan untuk pemrosesan acara menggunakan CloudWatch Acara karena aturan berusaha untuk mencocokkan pola dalam objek JSON. Untuk informasi selengkapnya, lihat Peristiwa dan pola acara serta peristiwa EMR Amazon di Panduan Pengguna CloudWatch Acara Amazon.

> **Note**
>
> Untuk memastikan bahwa kami memberi Anda informasi yang paling relevan, kami terus menyempurnakan pesan kesalahan kami. Oleh karena itu, kami menyarankan agar Anda tidak mengurai teks dari pesan untuk memulai tindakan selanjutnya dalam alur kerja Anda.

Acara awal cluster

Status atau perubahan status	Kepelikan	Tipe peristiwa	Kode acara	Pesan
`CREATING`	`WARN`	Penyediaan armada instans EMR Amazon	Penyediaan EC2 - Kapasitas Instans Tidak Cukup	Kami tidak dapat membuat klaster EMR Amazon Anda untuk Armada Instans Amazon `InstanceFleetID` EC2 memiliki kapasitas Spot yang tidak mencukupi `ClusterId (ClusterName)` untuk jenis Instans dan kapasitas Sesuai Permintaan yang tidak mencukupi untuk `[Instancetype1, Instancetype2]` jenis

Status atau perubahan status	Kepelikan	Tipe peristiwa	Kode acara	Pesan
				Instans di Availability Zone. `[Instance type3, Instancetype4] [AvailabilityZone1, AvaliabilityZone2]` Lihat dokumentasi di sini untuk informasi lebih lanjut tentang cara menanggapi acara ini.

Status atau perubahan status	Kepelikan	Tipe peristiwa	Kode acara	Pesan
`CREATING`	`WARN`	Penyediaan grup instans EMR Amazon	Penyediaan EC2 - Kapasitas Instans Tidak Cukup	Kami tidak dapat membuat klaster EMR Amazon Anda untuk Grup Instans Amazon `InstanceGroupID` EC2 memiliki kapasitas yang `[Spot or On-Demand]` tidak mencukupi `ClusterId (ClusterName)` untuk `InstanceType` jenis Instans di Availability Zone. `AvailabilityZone` Lihat dokumentasi di sini untuk informasi lebih lanjut tentang cara menanggapi acara ini.

Status atau perubahan status	Kepelikan	Tipe peristiwa	Kode acara	Pesan
`STARTING`	`INFO`	Perubahan status cluster EMR	tidak ada	Cluster Amazon EMR `ClusterId (ClusterName)` diminta `Time` dan sedang dibuat.

Status atau perubahan status	Kepelikan	Tipe peristiwa	Kode acara	Pesan
`STARTING`	`INFO`	Perubahan status cluster EMR	tidak ada	**Note** Hanya berlaku untuk cluster dengan konfigurasi armada instans dan beberapa Availability Zone yang dipilih dalam Amazon EC2. Cluster EMR Amazon `ClusterId` `(ClusterName)` sedang dibuat di zone `(AvailabilityZoneID )`, yang dipilih dari opsi Availabil

Status atau perubahan status	Kepelikan	Tipe peristiwa	Kode acara	Pesan
				ity Zone yang ditentukan.
`STARTING`	`INFO`	Perubahan status cluster EMR	tidak ada	Cluster Amazon EMR `ClusterId (ClusterName)` mulai menjalankan langkah-langkah di. `Time`

Status atau perubahan status	Kepelikan	Tipe peristiwa	Kode acara	Pesan
`WAITING`	`INFO`	Perubahan status cluster EMR	tidak ada	Cluster Amazon EMR `ClusterId (ClusterName)` dibuat di `Time` dan siap digunakan. - atau - Cluster Amazon EMR `ClusterId (ClusterName)` selesai menjalankan semua langkah yang tertunda di. `Time` **Note** Sebuah cluster di `WAITING` negara bagian mungkin masih memproses pekerjaan.

> **Note**
>
> Peristiwa dengan kode peristiwa muncul `EC2 provisioning - Insufficient Instance Capacity` secara berkala saat klaster EMR Anda mengalami kesalahan kapasitas yang tidak mencukupi dari Amazon EC2 untuk armada instans atau grup instans Anda selama operasi pembuatan atau pengubahan ukuran klaster. Untuk informasi tentang cara menanggapi peristiwa ini, lihatMenanggapi peristiwa kapasitas instans Amazon EMR cluster yang tidak mencukupi.

Peristiwa penghentian klaster

Status atau perubahan status	Kepelikan	Tipe peristiwa	Kode acara	Pesan
`TERMINATED`	Tingkat kepelikan tergantung pada alasan perubahan status, seperti yang ditunjukkan pada hal berikut: • **`CRITICAL`** jika klaster diakhiri dengan salah satu alasan perubahan status berikut: `INTERNAL_ERROR` , `VALIDATION_ERROR` , `INSTANCE_`	Perubahan status cluster EMR	tidak ada	Amazon EMR Cluster `ClusterId (ClusterName)` telah dihentikan pada `Time` dengan alasan. `StateChangeReason: Code`

Status atau perubahan status	Kepelikan	Tipe peristiwa	Kode acara	Pesan
	`FAILURE` , `BOOTSTRAP_FAILURE` , atau `STEP_FAILURE` . • **`INFO`** jika klaster diakhiri dengan salah satu alasan perubahan status berikut: `USER_REQUEST` atau `ALL_STEPS_COMPLETED` .			
`TERMINATED_WITH_ERRORS`	`CRITICAL`	Perubahan status cluster EMR	tidak ada	Amazon EMR Cluster `ClusterId (ClusterName)` telah diakhiri dengan kesalahan di `Time` dengan alasan. `StateChangeReason: Code`

Instance peristiwa perubahan negara armada

> **Note**
>
> Konfigurasi armada instance hanya tersedia di Amazon EMR rilis 4.8.0 dan yang lebih baru, tidak termasuk 5.0.0 dan 5.0.3.

Status atau perubahan status	Kepelikan	Tipe peristiwa	Kode acara	Pesan
Dari `PROVISIONING` ke `WAITING`	`INFO`		tidak ada	Penyediaan misalnya armada di `InstanceFleetID` klaster EMR Amazon selesai. `ClusterId (ClusterName)` Penyediaan dimulai pada `Time` dan memakan `Num` waktu beberapa menit. Armada instans sekarang memiliki kapasitas On-Demand `Num` dan kapasitas Spot. `Num` Kapasitas Target On-Demand adalah`Num`, dan kapasitas

Status atau perubahan status	Kepelikan	Tipe peristiwa	Kode acara	Pesan
				Spot target adalahNum.
Dari WAITING ke RESIZING	INFO		tidak ada	Pengubahan ukuran armada misalnya InstanceFleetID di ClusterId (ClusterName) klaster EMR Amazon dimulai pada. Time Armada instans mengubah ukuran dari kapasitas On-Demand Num ke targetNum, dan dari kapasitas Spot Num ke target. Num

Status atau perubahan status	Kepelikan	Tipe peristiwa	Kode acara	Pesan
Dari `RESIZING` ke `WAITING`	`INFO`		tidak ada	Operasi pengubahan ukuran untuk armada misalnya `InstanceFleetID` di `ClusterId (ClusterName)` cluster EMR Amazon selesai. Pengubahan ukuran dimulai pada `Time` dan memakan waktu `Num` beberapa menit. Armada instans sekarang memiliki kapasitas On-Demand `Num` dan kapasitas Spot. `Num` Kapasitas Target On-Demand adalah `Num` dan kapasitas Spot target adalah`Num`.

Status atau perubahan status	Kepelikan	Tipe peristiwa	Kode acara	Pesan
Dari `RESIZING` ke `WAITING`	`INFO`		tidak ada	Operasi pengubahan ukuran misalnya armada `InstanceFleetID` di `ClusterId` `(ClusterName)` cluster EMR Amazon telah mencapai batas waktu dan berhenti. Pengubahan ukuran dimulai pada `Time` dan berhenti setelah `Num` beberapa menit. Armada instans sekarang memiliki kapasitas On-Demand `Num` dan kapasitas Spot. `Num` Kapasitas Target On-Demand adalah `Num` dan kapasitas Spot target adalah`Num`.

Status atau perubahan status	Kepelikan	Tipe peristiwa	Kode acara	Pesan
SUSPENDED	ERROR		tidak ada	Armada instans InstanceFleetID di cluster EMR Amazon ClusterId (ClusterName) ditangkap karena Time alasan berikut:. ReasonDesc
RESIZING	WARNING		tidak ada	Operasi pengubahan ukuran untuk armada misalnya InstanceFleetID di ClusterId (ClusterName) cluster EMR Amazon macet karena alasan berikut:. ReasonDesc

Status atau perubahan status	Kepelikan	Tipe peristiwa	Kode acara	Pesan
`WAITING` atau `Running`	`INFO`		tidak ada	Operasi pengubahan ukuran misalnya armada `InstanceFleetID` di `ClusterId` `(ClusterName)` klaster EMR Amazon tidak dapat diselesaikan sementara Amazon EMR menambahkan kapasitas Spot di zona ketersediaan. `AvailabilityZone` Kami telah membatalkan permintaan Anda untuk menyediakan kapasitas Spot tambahan. Untuk tindakan yang disarankan, periksa Praktik terbaik misalnya dan fleksibilitas Availability Zone dan coba lagi.

Status atau perubahan status	Kepelikan	Tipe peristiwa	Kode acara	Pesan
`WAITING` atau `Running`	`INFO`		tidak ada	Operasi pengubahan ukuran untuk armada misalnya `InstanceFleetID` di `ClusterId` `(ClusterName)` cluster EMR Amazon dimulai `Entity` oleh at. `Time`

Acara mengubah ukuran armada instans

Tipe peristiwa	Kepelikan	Kode acara	Pesan
Pengubahan ukuran armada instans EMR Amazon	`ERROR`	Batas Waktu Penyediaan Spot	Operasi Resize untuk Instance Fleet `InstanceFleetID` di Amazon EMR `ClusterId` `(ClusterName)` cluster tidak dapat diselesaikan saat memperoleh kapasitas Spot di AZ. `AvailabilityZone` Kami sekarang telah membatalkan permintaan Anda dan

Tipe peristiwa	Kepelikan	Kode acara	Pesan
			berhenti mencoba menyediakan kapasitas Spot tambahan dan Armada Instance telah menyediak an kapasitas Spot sebesar. num Kapasitas Spot Target adalahnum. Untuk informasi lebih lanjut dan tindakan yang disarankan, silakan periksa halaman dokumentasi di sini dan coba lagi.

Tipe peristiwa	Kepelikan	Kode acara	Pesan
Pengubahan ukuran armada instans EMR Amazon	ERROR	Batas Waktu Penyediaan Sesuai Permintaan	Operasi Resize untuk Instance Fleet `InstanceFleetID` di Amazon EMR `ClusterId (ClusterName)` cluster tidak dapat diselesaikan saat memperoleh kapasitas On-Demand di AZ. `AvailabilityZone` Kami sekarang telah membatalkan permintaan Anda dan berhenti mencoba menyediakan kapasitas On-Demand tambahan dan Armada Instance telah menyediakan kapasitas On-Demand sebesar. `num` Kapasitas Target On-Demand adalah`num`. Untuk informasi lebih lanjut dan tindakan yang disarankan, silakan periksa halaman dokumentasi di sini dan coba lagi.

Tipe peristiwa	Kepelikan	Kode acara	Pesan
Pengubahan ukuran armada instans EMR Amazon	`WARNING`	Penyediaan EC2 - Kapasitas Instans Tidak Cukup	Kami tidak dapat menyelesaikan operasi pengubahan ukuran untuk Armada Instance `InstanceFleetID` di `ClusterId (ClusterName)` klaster EMR karena Amazon EC2 memiliki kapasitas Spot yang tidak mencukupi untuk jenis Instans dan kapasitas Sesuai Permintaan yang tidak mencukupi untuk `[Instancetype1, Instancetype2]` tipe Instans di Availability Zone. `[Instancetype3, Instancetype4] [AvailabilityZone1]` Sejauh ini, armada instans telah menyediakan kapasitas On-Demand `num` dan target kapasitas On-Demand adalah. `num` Kapasitas Spot yang disediakan adalah `num` dan

Tipe peristiwa	Kepelikan	Kode acara	Pesan
			kapasitas Spot target adalah. num Lihat dokumentasi di sini untuk informasi lebih lanjut tentang cara menanggapi acara ini.

Tipe peristiwa	Kepelikan	Kode acara	Pesan
Pengubahan ukuran armada instans EMR Amazon	`WARNING`	Batas Waktu Penyediaan Spot - Mengubah Ukuran Terus	Kami masih menyediakan kapasitas Spot untuk operasi pengubahan ukuran Armada Instance yang dimulai pada `time` misalnya ID armada di cluster EMR `InstanceFleetID` Amazon untuk di AZ. `ClusterId (ClusterName) [Instancetype1, Instancetype2] AvailabilityZone` Untuk operasi pengubahan ukuran sebelumnya yang dimulai pada`time`, periode batas waktu berakhir, sehingga Amazon EMR menghentikan penyediaan kapasitas Spot setelah menambahkan `num` instans yang diminta ke armada instans Anda. `num` Untuk informasi lebih lanjut, silakan periksa

Tipe peristiwa	Kepelikan	Kode acara	Pesan
			halaman dokumentasi di sini.

Tipe peristiwa	Kepelikan	Kode acara	Pesan
Pengubahan ukuran armada instans EMR Amazon	`WARNING`	Batas Waktu Penyediaan Sesuai Permintaan - Mengubah Ukuran Terus	Kami masih menyediakan kapasitas Sesuai Permintaan untuk operasi pengubaha n ukuran Armada Instance yang dimulai pada misalnya ID armada di `time` klaster EMR `InstanceFleetID` Amazon untuk di AZ. `ClusterId (ClusterN ame) [Instance type1, In stancetyp e2] AvailabilityZone` Untuk operasi pengubaha n ukuran sebelumny a yang dimulai pada`time`, periode batas waktu berakhir, sehingga Amazon EMR berhenti menyediakan kapasitas Sesuai Permintaan setelah menambahkan `num` instans yang diminta ke armada instans Anda. `num` Untuk informasi lebih

Tipe peristiwa	Kepelikan	Kode acara	Pesan
			lanjut, silakan periksa halaman dokumentasi di sini.

> **Note**
>
> Peristiwa batas waktu penyediaan dipancarkan saat Amazon EMR berhenti menyediakan kapasitas Spot atau On-Demand untuk armada setelah batas waktu berakhir. Untuk informasi tentang cara menanggapi peristiwa ini, lihatMenanggapi peristiwa batas waktu tunggu armada pengubahan ukuran armada cluster Amazon EMR.

Peristiwa grup instans

Tipe peristiwa	Kepelikan	Kode acara	Pesan
Dari `RESIZING` ke `Running`	`INFO`	tidak ada	Operasi pengubahan ukuran untuk grup instans `InstanceGroupID` di `ClusterId (ClusterName)` klaster EMR Amazon selesai. Sekarang memiliki hitungan instanceNum. Pengubahan ukuran dimulai pada `Time` dan membutuhkan waktu `Num` beberapa menit untuk menyelesaikannya.
Dari `RUNNING` ke `RESIZING`	`INFO`	tidak ada	Pengubahan ukuran untuk grup instans

Tipe peristiwa	Kepelikan	Kode acara	Pesan
			`InstanceGroupID` di `ClusterId` `(ClusterName)` klaster EMR Amazon dimulai pada. `Time` Ini mengubah ukuran dari hitungan instance `Num` ke`Num`.
`SUSPENDED`	`ERROR`	tidak ada	Grup instans `InstanceGroupID` di cluster EMR Amazon `ClusterId` `(ClusterName)` ditangkap karena `Time` alasan berikut:. `ReasonDesc`
`RESIZING`	`WARNING`	tidak ada	Operasi pengubahan ukuran untuk grup instans `InstanceGroupID` di `ClusterId` `(ClusterName)` klaster EMR Amazon macet karena alasan berikut:. `ReasonDesc`

Tipe peristiwa	Kepelikan	Kode acara	Pesan
Ubah ukuran grup instans EMR Amazon	`WARNING`	Penyediaan EC2 - Kapasitas Instans Tidak Cukup	Kami tidak dapat menyelesaikan operasi pengubahan ukuran yang dimulai pada Grup Instance di `ClusterId (ClusterName)` klaster EMR `time` karena Amazon EC2 `Spot/On Demand` tidak memiliki kapasitas yang cukup untuk jenis Instans `InstanceGroupID [Instance type]` di Availability Zone. `[AvailabilityZone1]` Sejauh ini, grup instance memiliki jumlah instance berjalan `num` dan jumlah instance yang diminta adalah`num`. Lihat dokumentasi di sini untuk informasi lebih lanjut tentang cara menanggapi acara ini.

Tipe peristiwa	Kepelikan	Kode acara	Pesan
Dari `RUNNING` ke `RESIZING`	`INFO`	tidak ada	Pengubahan ukuran untuk grup instans `InstanceGroupID` di `ClusterId (ClusterName)` klaster EMR Amazon dimulai `Entity` oleh at. `Time`

> **Note**
>
> Dengan Amazon EMR versi 5.21.0 dan yang lebih baru, Anda dapat mengganti konfigurasi klaster dan menentukan klasifikasi konfigurasi tambahan untuk setiap grup instans dalam klaster berjalan. Anda dapat melakukan ini dengan menggunakan konsol Amazon EMR, AWS Command Line Interface (AWS CLI), atau AWS SDK. Untuk informasi lebih lanjut, lihat Menyediakan Konfigurasi untuk Grup Instans dalam Klaster Berjalan.

Tabel berikut mencantumkan peristiwa Amazon EMR, untuk operasi konfigurasi ulang, bersama dengan status atau perubahan status yang ditunjukkan oleh peristiwa tersebut, kepelikan peristiwa, dan pesan peristiwa.

Status atau perubahan status	Kepelikan	Pesan
`RUNNING`	`INFO`	Konfigurasi ulang untuk grup instans `InstanceGroupID` di `ClusterId (ClusterName)` kluster EMR Amazon dimulai oleh pengguna di. `Time` Versi konfigurasi yang diminta adalahNum.
Dari `RECONFIGURING` ke `Running`	`INFO`	Operasi konfigurasi ulang untuk grup instans

Status atau perubahan status	Kepelikan	Pesan
		`InstanceGroupID` di `ClusterId (ClusterName)` kluster EMR Amazon selesai. Konfigurasi ulang dimulai pada `Time` dan membutuhkan waktu `Num` beberapa menit untuk menyelesaikannya. Versi konfigurasi saat ini adalah`Num`.
Dari `RUNNING` ke `RECONFIGURING` in	`INFO`	Konfigurasi ulang untuk grup instans `InstanceGroupID` di `ClusterId (ClusterName)` kluster EMR Amazon dimulai pada. `Time` Ini mengkonfigurasi dari nomor versi `Num` ke nomor `Num` versi.
`RESIZING`	`INFO`	Mengkonfigurasi ulang operasi ke versi konfigurasi `Num` untuk grup instans `InstanceGroupID` di klaster `ClusterId (ClusterName)` EMR Amazon diblokir sementara `Time` karena grup instans masuk. `State`

Status atau perubahan status	Kepelikan	Pesan
RECONFIGURING	INFO	Mengubah ukuran operasi menuju jumlah instans Num untuk grup instans InstanceGroupID di ClusterId (ClusterName) klaster EMR Amazon diblokir Time sementara karena grup instans masuk. State
RECONFIGURING	WARNING	Operasi konfigurasi ulang untuk grup instans InstanceGroupID di ClusterId (ClusterName) kluster EMR Amazon gagal Time dan Num membutuhkan waktu beberapa menit untuk gagal. Versi konfigurasi yang gagal adalahNum.
RECONFIGURING	INFO	Konfigurasi kembali ke nomor versi sukses sebelumnya Num untuk grup instans InstanceGroupID di cluster EMR Amazon di. ClusterId (ClusterName) Time Versi konfigurasi baru adalahNum.

Status atau perubahan status	Kepelikan	Pesan
Dari RECONFIGURING ke Running	INFO	Konfigurasi berhasil dikembalikan ke versi sukses sebelumnya Num untuk grup instans InstanceGroupID di cluster EMR Amazon di. ClusterId (ClusterName) Time Versi konfigurasi baru adalahNum.
Dari RECONFIGURING ke SUSPENDED	CRITICAL	Gagal mengembalikan ke versi sukses sebelumnya Num untuk grup Instans InstanceGroupID di klaster EMR ClusterId (ClusterName) Amazon di. Time

Peristiwa kebijakan penskalaan otomatis

Status atau perubahan status	Kepelikan	Pesan
PENDING	INFO	Kebijakan Auto Scaling ditambahkan ke grup instans di klaster EMR InstanceGroupID Amazon di. ClusterId (ClusterName) Time Kebijakan ini lampiran tertunda. - atau - Kebijakan Auto Scaling untuk grup instans di klaster ClusterId (ClusterName) EMR InstanceG

Status atau perubahan status	Kepelikan	Pesan
		roupID Amazon telah diperbarui di. Time Kebijakan ini lampiran tertunda.
ATTACHED	INFO	Kebijakan Auto Scaling untuk grup instans di klaster ClusterId (Cluster Name) EMR InstanceG roupID Amazon dilampirkan pada. Time
DETACHED	INFO	Kebijakan Auto Scaling untuk grup instans di klaster ClusterId (Cluster Name) EMR InstanceG roupID Amazon terlepas di. Time
FAILED	ERROR	Kebijakan Auto Scaling untuk grup instans di klaster EMR InstanceGroupID Amazon tidak ClusterId (ClusterName) dapat dilampirkan dan gagal di. Time - atau - Kebijakan Auto Scaling untuk grup instans di klaster EMR InstanceGroupID Amazon tidak ClusterId (ClusterName) dapat dilepas dan gagal. Time

Peristiwa langkah

Status atau perubahan status	Kepelikan	Pesan
PENDING	INFO	Langkah StepID (StepName) telah ditambahkan ke Amazon EMR cluster ClusterId (ClusterName) di Time dan sedang menunggu eksekusi.
CANCEL_PENDING	WARN	Langkah StepID (StepName) di Amazon EMR cluster dibatalkan pada Time dan ClusterId (ClusterName) sedang menunggu pembatalan.
RUNNING	INFO	Langkah StepID (StepName) di Amazon EMR cluster ClusterId (ClusterName) mulai berjalan di. Time
COMPLETED	INFO	Langkah StepID (StepName) di Amazon EMR cluster ClusterId (ClusterName) menyelesaikan eksekusi di. Time Langkah mulai berjalan Time dan membutuhkan waktu Num beberapa menit untuk menyelesaikannya.
CANCELLED	WARN	Permintaan pembatalan telah berhasil untuk langkah klaster di klaster EMR

Status atau perubahan status	Kepelikan	Pesan
		`StepID (StepName)` Amazon `ClusterId (ClusterName)` di`Time`, dan langkahnya sekarang dibatalkan.
`FAILED`	`ERROR`	Langkah `StepID (StepName)` di Amazon EMR cluster `ClusterId (ClusterName)` gagal di. `Time`

Melihat acara dengan konsol EMR Amazon

Untuk setiap klaster, Anda dapat melihat daftar sederhana dari peristiwa di panel detail, yang berisi daftar peristiwa dalam urutan kejadian. Anda juga dapat melihat semua peristiwa untuk semua klaster dalam suatu wilayah dalam urutan kejadian.

Jika Anda tidak ingin pengguna melihat semua peristiwa klaster untuk suatu wilayah, tambahkan pernyataan yang menolak izin (`"Effect": "Deny"`) untuk tindakan `elasticmapreduce:ViewEventsFromAllClustersInConsole` ke kebijakan yang melekat pada pengguna tersebut.

> **Note**
>
> Kami telah mendesain ulang konsol EMR Amazon agar lebih mudah digunakan. Lihat Apa yang baru dengan konsol? untuk mempelajari tentang perbedaan antara pengalaman konsol lama dan baru.

New console

Untuk melihat peristiwa untuk semua cluster di Wilayah dengan konsol baru

1. Masuk keAWS Management Console, dan buka konsol EMR Amazon di https://console.aws.amazon.com/emr.

2. Di bawah EMR pada EC2 di panel navigasi kiri, pilih Acara.

Untuk melihat peristiwa untuk klaster tertentu dengan konsol baru

1. Masuk keAWS Management Console, dan buka konsol EMR Amazon di https://console.aws.amazon.com/emr.
2. Di bawah EMR pada EC2 di panel navigasi kiri, pilih Clusters, lalu pilih cluster.
3. Untuk melihat semua acara Anda, pilih tab Acara di halaman detail klaster.

Old console

Untuk melihat peristiwa untuk semua cluster di Wilayah dengan konsol lama

1. Buka konsol Amazon EMR di https://console.aws.amazon.com/elasticmapreduce/.
2. Pilih Peristiwa.

Untuk melihat peristiwa untuk cluster tertentu dengan konsol lama

1. Buka konsol Amazon EMR di https://console.aws.amazon.com/elasticmapreduce/.
2. Pilih Daftar Klaster, pilih klaster, lalu pilih Lihat detail.
3. Pilih Peristiwa di panel detail klaster.

Menanggapi peristiwa CloudWatch

Bagian ini menjelaskan berbagai cara agar Anda dapat menanggapi peristiwa yang dapat ditindaklanjuti yang dipancarkan Amazon EMR sebagai pesan acara. CloudWatch

Topik

- Membuat aturan untuk acara EMR Amazon dengan CloudWatch
- Menyetel alarm pada metrik CloudWatch
- Menanggapi peristiwa kapasitas instans Amazon EMR cluster yang tidak mencukupi
- Menanggapi peristiwa batas waktu tunggu armada pengubahan ukuran armada cluster Amazon EMR

Membuat aturan untuk acara EMR Amazon dengan CloudWatch

Amazon EMR secara otomatis mengirimkan acara ke aliran CloudWatch acara. Anda dapat membuat aturan yang mencocokan peristiwa ke pola tertentu, dan merutekan peristiwa ke target untuk mengambil tindakan, seperti mengirim notifikasi email. Pola dicocokkan terhadap peristiwa objek JSON. Untuk informasi selengkapnya tentang detail acara EMR Amazon, lihat peristiwa EMR Amazon di Panduan Pengguna Acara Amazon CloudWatch .

Untuk informasi tentang menyiapkan aturan CloudWatch acara, lihat Membuat CloudWatch aturan yang memicu peristiwa.

Menyetel alarm pada metrik CloudWatch

Amazon EMR mendorong metrik ke Amazon. CloudWatch Sebagai tanggapan, Anda dapat menggunakan CloudWatch untuk menyetel alarm pada metrik Amazon EMR Anda. Misalnya, Anda dapat mengonfigurasi alarm CloudWatch untuk mengirimi Anda email kapan saja penggunaan HDFS naik di atas 80%. Untuk petunjuk terperinci, lihat Membuat atau mengedit CloudWatch alarm di Panduan CloudWatch Pengguna Amazon.

Menanggapi peristiwa kapasitas instans Amazon EMR cluster yang tidak mencukupi

Gambaran Umum

Cluster EMR Amazon mengembalikan kode peristiwa `EC2 provisioning - Insufficient Instance Capacity` ketika Availability Zone yang dipilih tidak memiliki kapasitas yang cukup untuk memenuhi permintaan awal atau pengubahan ukuran klaster Anda. Peristiwa akan muncul secara berkala dengan grup instans dan armada instans jika EMR Amazon berulang kali menemukan pengecualian kapasitas yang tidak mencukupi dan tidak dapat memenuhi permintaan penyediaan Anda untuk operasi pengaktifan klaster atau pengubahan ukuran klaster.

Halaman ini menjelaskan cara terbaik Anda merespons jenis peristiwa ini saat terjadi untuk klaster EMR Anda.

Respons yang disarankan untuk acara kapasitas yang tidak mencukupi

Kami menyarankan Anda menanggapi peristiwa kapasitas yang tidak memadai dengan salah satu cara berikut:

- Tunggu kapasitas untuk pulih. Kapasitas sering bergeser, sehingga pengecualian kapasitas yang tidak mencukupi dapat pulih dengan sendirinya. Cluster Anda akan mulai atau selesai mengubah ukuran segera setelah kapasitas Amazon EC2 tersedia.

- Atau, Anda dapat menghentikan klaster, memodifikasi konfigurasi tipe instans, dan membuat klaster baru dengan permintaan konfigurasi klaster yang diperbarui. Untuk informasi selengkapnya, lihat Praktik terbaik misalnya dan fleksibilitas Availability Zone.

Anda juga dapat mengatur aturan atau respons otomatis terhadap peristiwa kapasitas yang tidak mencukupi, seperti yang dijelaskan di bagian berikutnya.

Pemulihan otomatis dari peristiwa kapasitas yang tidak mencukupi

Anda dapat membangun otomatisasi dalam menanggapi peristiwa EMR Amazon seperti yang memiliki kode acara. `EC2 provisioning - Insufficient Instance Capacity` Misalnya, AWS Lambda fungsi berikut mengakhiri klaster EMR dengan grup instans yang menggunakan instance On-Demand, dan kemudian membuat klaster EMR baru dengan grup instans yang berisi tipe instans yang berbeda dari permintaan asli.

Kondisi berikut memicu proses otomatis terjadi:

- Peristiwa kapasitas yang tidak mencukupi telah dipancarkan untuk node primer atau inti selama lebih dari 20 menit.
- Cluster tidak dalam keadaan READY atau WAITING. Untuk informasi lebih lanjut tentang status klaster EMR, lihat. Memahami siklus hidup klaster

> **Note**
>
> Ketika Anda membangun proses otomatis untuk pengecualian kapasitas yang tidak mencukupi, Anda harus mempertimbangkan bahwa peristiwa kapasitas yang tidak mencukupi dapat dipulihkan. Kapasitas sering bergeser dan cluster Anda akan melanjutkan pengubahan ukuran atau mulai beroperasi segera setelah kapasitas Amazon EC2 tersedia.

Example berfungsi untuk menanggapi peristiwa kapasitas yang tidak mencukupi

```
// Lambda code with Python 3.10 and handler is lambda_function.lambda_handler
// Note: related IAM role requires permission to use Amazon EMR

import json
import boto3
import datetime
from datetime import timezone
```

```
INSUFFICIENT_CAPACITY_EXCEPTION_DETAIL_TYPE = "EMR Instance Group Provisioning"
INSUFFICIENT_CAPACITY_EXCEPTION_EVENT_CODE = (
    "EC2 provisioning - Insufficient Instance Capacity"
)
ALLOWED_INSTANCE_TYPES_TO_USE = [
    "m5.xlarge",
    "c5.xlarge",
    "m5.4xlarge",
    "m5.2xlarge",
    "t3.xlarge",
]
CLUSTER_START_ACCEPTABLE_STATES = ["WAITING", "RUNNING"]
CLUSTER_START_SLA = 20

CLIENT = boto3.client("emr", region_name="us-east-1")

# checks if the incoming event is 'EMR Instance Fleet Provisioning' with eventCode 'EC2
 provisioning - Insufficient Instance Capacity'
def is_insufficient_capacity_event(event):
    if not event["detail"]:
        return False
    else:
        return (
            event["detail-type"] == INSUFFICIENT_CAPACITY_EXCEPTION_DETAIL_TYPE
            and event["detail"]["eventCode"]
            == INSUFFICIENT_CAPACITY_EXCEPTION_EVENT_CODE
        )


# checks if the cluster is eligible for termination
def is_cluster_eligible_for_termination(event, describeClusterResponse):
    # instanceGroupType could be CORE, MASTER OR TASK
    instanceGroupType = event["detail"]["instanceGroupType"]
    clusterCreationTime = describeClusterResponse["Cluster"]["Status"]["Timeline"][
        "CreationDateTime"
    ]
    clusterState = describeClusterResponse["Cluster"]["Status"]["State"]

    now = datetime.datetime.now()
    now = now.replace(tzinfo=timezone.utc)
    isClusterStartSlaBreached = clusterCreationTime < now - datetime.timedelta(
        minutes=CLUSTER_START_SLA
    )
```

```
    # Check if instance group receiving Insufficient capacity exception is CORE or
 PRIMARY (MASTER),
    # and it's been more than 20 minutes since cluster was created but the cluster
 state and the cluster state is not updated to RUNNING or WAITING
    if (
        (instanceGroupType == "CORE" or instanceGroupType == "MASTER")
        and isClusterStartSlaBreached
        and clusterState not in CLUSTER_START_ACCEPTABLE_STATES
    ):
        return True
    else:
        return False


# Choose item from the list except the exempt value
def choice_excluding(exempt):
    for i in ALLOWED_INSTANCE_TYPES_TO_USE:
        if i != exempt:
            return i


# Create a new cluster by choosing different InstanceType.
def create_cluster(event):
    # instanceGroupType cloud be CORE, MASTER OR TASK
    instanceGroupType = event["detail"]["instanceGroupType"]

    # Following two lines assumes that the customer that created the cluster already
 knows which instance types they use in original request
    instanceTypesFromOriginalRequestMaster = "m5.xlarge"
    instanceTypesFromOriginalRequestCore = "m5.xlarge"

    # Select new instance types to include in the new createCluster request
    instanceTypeForMaster = (
        instanceTypesFromOriginalRequestMaster
        if instanceGroupType != "MASTER"
        else choice_excluding(instanceTypesFromOriginalRequestMaster)
    )
    instanceTypeForCore = (
        instanceTypesFromOriginalRequestCore
        if instanceGroupType != "CORE"
        else choice_excluding(instanceTypesFromOriginalRequestCore)
    )
```

```
    print("Starting to create cluster...")
    instances = {
        "InstanceGroups": [
            {
                "InstanceRole": "MASTER",
                "InstanceCount": 1,
                "InstanceType": instanceTypeForMaster,
                "Market": "ON_DEMAND",
                "Name": "Master",
            },
            {
                "InstanceRole": "CORE",
                "InstanceCount": 1,
                "InstanceType": instanceTypeForCore,
                "Market": "ON_DEMAND",
                "Name": "Core",
            },
        ]
    }
    response = CLIENT.run_job_flow(
        Name="Test Cluster",
        Instances=instances,
        VisibleToAllUsers=True,
        JobFlowRole="EMR_EC2_DefaultRole",
        ServiceRole="EMR_DefaultRole",
        ReleaseLabel="emr-6.10.0",
    )

    return response["JobFlowId"]


# Terminated the cluster using clusterId received in an event
def terminate_cluster(event):
    print("Trying to terminate cluster, clusterId: " + event["detail"]["clusterId"])
    response = CLIENT.terminate_job_flows(JobFlowIds=[event["detail"]["clusterId"]])
    print(f"Terminate cluster response: {response}")


def describe_cluster(event):
    response = CLIENT.describe_cluster(ClusterId=event["detail"]["clusterId"])
    return response


def lambda_handler(event, context):
```

```
    if is_insufficient_capacity_event(event):
        print(
            "Received insufficient capacity event for instanceGroup, clusterId: "
            + event["detail"]["clusterId"]
        )

        describeClusterResponse = describe_cluster(event)

        shouldTerminateCluster = is_cluster_eligible_for_termination(
            event, describeClusterResponse
        )
        if shouldTerminateCluster:
            terminate_cluster(event)

            clusterId = create_cluster(event)
            print("Created a new cluster, clusterId: " + clusterId)
        else:
            print(
                "Cluster is not eligible for termination, clusterId: "
                + event["detail"]["clusterId"]
            )

    else:
        print("Received event is not insufficient capacity event, skipping")
```

Menanggapi peristiwa batas waktu tunggu armada pengubahan ukuran armada cluster Amazon EMR

Gambaran Umum

Cluster EMR Amazon memancarkan peristiwa saat menjalankan operasi pengubahan ukuran untuk cluster armada misalnya. Peristiwa batas waktu penyediaan dipancarkan saat Amazon EMR berhenti menyediakan kapasitas Spot atau On-Demand untuk armada setelah batas waktu berakhir. Durasi batas waktu dapat dikonfigurasi oleh pengguna sebagai bagian dari spesifikasi pengubahan ukuran untuk armada instance. Dalam skenario pengubahan ukuran berturut-turut untuk armada instans yang sama, Amazon EMR memancarkan `Spot provisioning timeout - continuing resize` atau `On-Demand provisioning timeout - continuing resize` peristiwa saat batas waktu untuk operasi pengubahan ukuran saat ini kedaluwarsa. Kemudian mulai menyediakan kapasitas untuk operasi pengubahan ukuran armada berikutnya.

Menanggapi peristiwa timeout pengubahan ukuran armada instance

Kami menyarankan Anda menanggapi peristiwa batas waktu penyediaan dengan salah satu cara berikut:

- Kunjungi kembali spesifikasi pengubahan ukuran dan coba lagi operasi pengubahan ukuran. Karena kapasitas sering bergeser, cluster Anda akan berhasil mengubah ukuran segera setelah kapasitas Amazon EC2 tersedia. Kami menyarankan pelanggan untuk mengonfigurasi nilai yang lebih rendah untuk durasi waktu tunggu untuk pekerjaan yang membutuhkan SLA yang lebih ketat.
- Atau, Anda dapat:
  - Luncurkan klaster baru dengan beragam jenis instans berdasarkan praktik terbaik misalnya dan fleksibilitas Availability Zone atau
  - Luncurkan cluster dengan kapasitas On-Demand
- Untuk waktu tunggu penyediaan - melanjutkan acara pengubahan ukuran, Anda juga dapat menunggu operasi pengubahan ukuran diproses. Amazon EMR akan terus memproses secara berurutan operasi pengubahan ukuran yang dipicu untuk armada, dengan menghormati spesifikasi pengubahan ukuran yang dikonfigurasi.

Anda juga dapat mengatur aturan atau tanggapan otomatis untuk acara ini seperti yang dijelaskan di bagian berikutnya.

Pemulihan otomatis dari peristiwa batas waktu penyediaan

Anda dapat membangun otomatisasi dalam menanggapi peristiwa EMR Amazon dengan kode `Spot Provisioning timeout` acara. Misalnya, AWS Lambda fungsi berikut mematikan klaster EMR dengan armada instans yang menggunakan instance Spot untuk node Tugas, dan kemudian membuat klaster EMR baru dengan armada instance yang berisi tipe instans yang lebih beragam daripada permintaan asli. Dalam contoh ini, `Spot Provisioning timeout` peristiwa yang dipancarkan untuk node tugas akan memicu eksekusi fungsi Lambda.

Example Contoh fungsi untuk menanggapi **`Spot Provisioning timeout`** peristiwa

```
// Lambda code with Python 3.10 and handler is lambda_function.lambda_handler
// Note: related IAM role requires permission to use Amazon EMR

import json
import boto3
import datetime
from datetime import timezone
```

```
SPOT_PROVISIONING_TIMEOUT_EXCEPTION_DETAIL_TYPE = "EMR Instance Fleet Resize"
SPOT_PROVISIONING_TIMEOUT_EXCEPTION_EVENT_CODE = (
    "Spot Provisioning timeout"
)

CLIENT = boto3.client("emr", region_name="us-east-1")

# checks if the incoming event is 'EMR Instance Fleet Resize' with eventCode 'Spot
 provisioning timeout'
def is_spot_provisioning_timeout_event(event):
    if not event["detail"]:
        return False
    else:
        return (
            event["detail-type"] == SPOT_PROVISIONING_TIMEOUT_EXCEPTION_DETAIL_TYPE
            and event["detail"]["eventCode"]
            == SPOT_PROVISIONING_TIMEOUT_EXCEPTION_EVENT_CODE
        )


# checks if the cluster is eligible for termination
def is_cluster_eligible_for_termination(event, describeClusterResponse):
    # instanceFleetType could be CORE, MASTER OR TASK
    instanceFleetType = event["detail"]["instanceFleetType"]

    # Check if instance fleet receiving Spot provisioning timeout event is TASK
    if (instanceFleetType == "TASK"):
        return True
    else:
        return False


# create a new cluster by choosing different InstanceType.
def create_cluster(event):
    # instanceFleetType cloud be CORE, MASTER OR TASK
    instanceFleetType = event["detail"]["instanceFleetType"]

    # the following two lines assumes that the customer that created the cluster
 already knows which instance types they use in original request
    instanceTypesFromOriginalRequestMaster = "m5.xlarge"
    instanceTypesFromOriginalRequestCore = "m5.xlarge"

    # select new instance types to include in the new createCluster request
```

```
    instanceTypesForTask = [
        "m5.xlarge",
        "m5.2xlarge",
        "m5.4xlarge",
        "m5.8xlarge",
        "m5.12xlarge"
    ]

    print("Starting to create cluster...")
    instances = {
        "InstanceFleets": [
            {
                "InstanceFleetType":"MASTER",
                "TargetOnDemandCapacity":1,
                "TargetSpotCapacity":0,
                "InstanceTypeConfigs":[
                    {
                        'InstanceType': instanceTypesFromOriginalRequestMaster,
                        "WeightedCapacity":1,
                    }
                ]
            },
            {
                "InstanceFleetType":"CORE",
                "TargetOnDemandCapacity":1,
                "TargetSpotCapacity":0,
                "InstanceTypeConfigs":[
                    {
                        'InstanceType': instanceTypesFromOriginalRequestCore,
                        "WeightedCapacity":1,
                    }
                ]
            },
            {
                "InstanceFleetType":"TASK",
                "TargetOnDemandCapacity":0,
                "TargetSpotCapacity":100,
                "LaunchSpecifications":{},
                "InstanceTypeConfigs":[
                    {
                        'InstanceType': instanceTypesForTask[0],
                        "WeightedCapacity":1,
                    },
                    {
```

```
                        'InstanceType': instanceTypesForTask[1],
                        "WeightedCapacity":2,
                    },
                    {
                        'InstanceType': instanceTypesForTask[2],
                        "WeightedCapacity":4,
                    },
                    {
                        'InstanceType': instanceTypesForTask[3],
                        "WeightedCapacity":8,
                    },
                    {
                        'InstanceType': instanceTypesForTask[4],
                        "WeightedCapacity":12,
                    }
                ],
                "ResizeSpecifications": {
                    "SpotResizeSpecification": {
                        "TimeoutDurationMinutes": 30
                    }
                }
            }
        ]
    }
    response = CLIENT.run_job_flow(
        Name="Test Cluster",
        Instances=instances,
        VisibleToAllUsers=True,
        JobFlowRole="EMR_EC2_DefaultRole",
        ServiceRole="EMR_DefaultRole",
        ReleaseLabel="emr-6.10.0",
    )

    return response["JobFlowId"]


# terminated the cluster using clusterId received in an event
def terminate_cluster(event):
    print("Trying to terminate cluster, clusterId: " + event["detail"]["clusterId"])
    response = CLIENT.terminate_job_flows(JobFlowIds=[event["detail"]["clusterId"]])
    print(f"Terminate cluster response: {response}")


def describe_cluster(event):
```

```
    response = CLIENT.describe_cluster(ClusterId=event["detail"]["clusterId"])
    return response


def lambda_handler(event, context):
    if is_spot_provisioning_timeout_event(event):
        print(
            "Received spot provisioning timeout event for instanceFleet, clusterId: "
            + event["detail"]["clusterId"]
        )

        describeClusterResponse = describe_cluster(event)

        shouldTerminateCluster = is_cluster_eligible_for_termination(
            event, describeClusterResponse
        )
        if shouldTerminateCluster:
            terminate_cluster(event)

            clusterId = create_cluster(event)
            print("Created a new cluster, clusterId: " + clusterId)
        else:
            print(
                "Cluster is not eligible for termination, clusterId: "
                + event["detail"]["clusterId"]
            )

    else:
        print("Received event is not spot provisioning timeout event, skipping")
```

Melihat metrik aplikasi klaster dengan Ganglia

Ganglia tersedia dengan Amazon EMR rilis 4.2 ke atas. Ganglia adalah proyek sumber terbuka yang merupakan sistem terdistribusi yang dapat diskalakan, yang dirancang untuk memantau klaster dan grid sekaligus meminimalisir dampak terhadap performanya. Ketika Anda mengaktifkan Ganglia di klaster Anda, Anda dapat menghasilkan laporan dan melihat performa klaster secara keseluruhan, serta memeriksa performa masing-masing instans simpul. Ganglia juga dikonfigurasi untuk menyerap dan memvisualisasikan metrik Hadoop dan Spark. Untuk informasi selengkapnya, lihat Ganglia dalam Panduan Rilis Amazon EMR.

Logging panggilan API Amazon EMR di AWS CloudTrail

Amazon EMR terintegrasi denganAWS CloudTrail, layanan yang menyediakan catatan tindakan yang diambil oleh pengguna, peran, atau AWS layanan di Amazon EMR. CloudTrail menangkap semua panggilan API untuk Amazon EMR sebagai acara. Panggilan yang direkam mencakup panggilan dari konsol Amazon EMR dan panggilan kode ke operasi API Amazon EMR. Jika Anda membuat jejak, Anda dapat mengaktifkan pengiriman CloudTrail acara secara terus menerus ke bucket Amazon S3, termasuk acara untuk Amazon EMR. 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 ke Amazon EMR, alamat IP dari mana permintaan dibuat, siapa yang membuat permintaan, kapan dibuat, dan detail tambahan.

Untuk mempelajari selengkapnya CloudTrail, lihat Panduan AWS CloudTrail Pengguna.

Informasi EMR Amazon di CloudTrail

CloudTrail diaktifkan di AWS akun Anda saat Anda membuat akun. Ketika aktivitas terjadi di Amazon EMR, aktivitas tersebut direkam dalam suatu CloudTrail peristiwa bersama dengan peristiwa AWS layanan lainnya dalam riwayat Acara. Anda dapat melihat, mencari, dan mengunduh peristiwa terbaru di akun AWS. Untuk informasi selengkapnya, lihat Melihat peristiwa dengan riwayat CloudTrail acara.

Untuk catatan peristiwa yang sedang berlangsung di akun AWS Anda, termasuk peristiwa untuk Amazon EMR, buatlah jejak. Jejak memungkinkan CloudTrail untuk mengirimkan file log ke bucket Amazon S3. Secara default, saat Anda membuat jejak di dalam konsol tersebut, jejak diterapkan ke semua Wilayah AWS. Jejak mencatat peristiwa dari semua Wilayah di partisi AWS dan mengirimkan file log ke bucket Amazon S3 yang Anda tentukan. Selain itu, Anda dapat mengonfigurasi 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 EMR Amazon dicatat oleh CloudTrail dan didokumentasikan dalam Referensi API EMR Amazon. Misalnya, panggilan ke`RunJobFlow`, `ListCluster` dan `DescribeCluster` tindakan menghasilkan entri dalam file CloudTrail log.

Setiap entri peristiwa atau log berisi informasi tentang siapa yang membuat permintaan tersebut. Informasi identitas membantu Anda menentukan berikut ini:

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

Dalam kasus di mana proses, bukan pengguna, membuat klaster, Anda dapat menggunakan `principalId` pengenal untuk menentukan pengguna yang terkait dengan pembuatan klaster. Untuk informasi selengkapnya, lihat Elemen userIdentity CloudTrail .

Contoh: entri berkas log Amazon EMR

Trail adalah konfigurasi yang memungkinkan pengiriman peristiwa sebagai file log ke bucket Amazon S3 yang Anda tentukan. CloudTrail file log berisi satu atau lebih entri log. Peristiwa mewakili permintaan tunggal dari sumber manapun dan mencakup informasi tentang tindakan yang diminta, tanggal dan waktu tindakan, parameter permintaan, dan sebagainya. CloudTrail file log bukanlah jejak tumpukan yang diurutkan dari panggilan API publik, jadi file tersebut tidak muncul dalam urutan tertentu.

Contoh berikut menunjukkan entri CloudTrail log yang menunjukkan RunJobFlowtindakan.

```
{
 "Records": [
 {
         "eventVersion":"1.01",
         "userIdentity":{
            "type":"IAMUser",
            "principalId":"EX_PRINCIPAL_ID",
            "arn":"arn:aws:iam::123456789012:user/temporary-user-xx-7M",
            "accountId":"123456789012",
            "userName":"temporary-user-xx-7M"
         },
```

```
         "eventTime":"2018-03-31T17:59:21Z",
         "eventSource":"elasticmapreduce.amazonaws.com",
         "eventName":"RunJobFlow",
         "awsRegion":"us-west-2",
         "sourceIPAddress":"192.0.2.1",
         "userAgent":"aws-sdk-java/unknown-version Linux/xx Java_HotSpot(TM)_64-
Bit_Server_VM/xx",
         "requestParameters":{
            "tags":[
               {
                  "value":"prod",
                  "key":"domain"
               },
               {
                  "value":"us-west-2",
                  "key":"realm"
               },
               {
                  "value":"VERIFICATION",
                  "key":"executionType"
               }
            ],
            "instances":{
               "slaveInstanceType":"m5.xlarge",
               "ec2KeyName":"emr-integtest",
               "instanceCount":1,
               "masterInstanceType":"m5.xlarge",
               "keepJobFlowAliveWhenNoSteps":true,
               "terminationProtected":false
            },
            "visibleToAllUsers":false,
            "name":"MyCluster",
            "ReleaseLabel":"emr-5.16.0"
         },
         "responseElements":{
            "jobFlowId":"j-2WDJCGEG4E6AJ"
         },
         "requestID":"2f482daf-b8fe-11e3-89e7-75a3d0e071c5",
         "eventID":"b348a38d-f744-4097-8b2a-e68c9b424698"
      },
  ...additional entries
   ]
}
```

Gunakan penskalaan cluster

Anda dapat menyesuaikan jumlah instans Amazon EC2 yang tersedia untuk klaster Amazon EMR secara otomatis atau manual dalam menanggapi beban kerja yang memiliki berbagai tuntutan. Untuk menggunakan penskalaan otomatis, Anda memiliki dua opsi. Anda dapat mengaktifkan penskalaan terkelola Amazon EMR atau membuat kebijakan penskalaan otomatis khusus. Tabel berikut menjelaskan perbedaan antara dua opsi tersebut.

	Penskalaan terkelola Amazon EMR	Penskalaan otomatis kustom
Kebijakan dan aturan penskalaan	Tidak ada kebijakan yang diperlukan. Amazon EMR mengelola aktivitas penskalaan otomatis dengan terus mengevaluasi metrik klaster dan membuat keputusan penskalaan yang dioptimalkan.	Anda perlu menentukan dan mengelola kebijakan dan aturan penskalaan otomatis, seperti kondisi spesifik yang memicu aktivitas penskalaan, periode evaluasi, periode pendinginan, dll.
Rilis Amazon EMR yang didukung	Amazon EMR versi 5.30.0 dan lebih tinggi (kecuali Amazon EMR versi 6.0.0)	Amazon EMR versi 4.0.0 dan lebih tinggi
Komposisi klaster yang didukung	Grup instans atau armada instans	Grup instans saja
Konfigurasi batas penskalaan	Batas penskalaan dikonfigurasi untuk seluruh klaster.	Batas penskalaan hanya dapat dikonfigurasi untuk setiap grup instans.
Frekuensi evaluasi metrik	Setiap 5 sampai 10 detik Evaluasi metrik yang lebih sering memungkinkan Amazon EMR membuat	Anda dapat menentukan periode evaluasi hanya dalam penambahan lima menit.

	Penskalaan terkelola Amazon EMR	Penskalaan otomatis kustom
	keputusan penskalaan yang lebih tepat.	
Aplikasi yang didukung	Hanya aplikasi YARN yang didukung, seperti Spark, Hadoop, Hive, Flink. Penskalaan terkelola Amazon EMR tidak mendukung aplikasi yang tidak didasarkan pada YARN, seperti Presto atau HBase.	Anda dapat memilih aplikasi mana yang didukung saat menentukan aturan penskalaan otomatis.

Pertimbangan-pertimbangan

- Cluster EMR Amazon selalu terdiri dari satu atau tiga node utama. Setelah Anda awalnya mengkonfigurasi cluster, Anda hanya dapat menskalakan inti dan node tugas. Anda tidak dapat menskalakan jumlah node utama untuk cluster.
- Misalnya grup, operasi konfigurasi ulang dan operasi pengubahan ukuran terjadi secara berurutan dan tidak bersamaan. Jika Anda memulai konfigurasi ulang saat grup instans mengubah ukuran, konfigurasi ulang dimulai setelah grup instance menyelesaikan pengubahan ukuran yang sedang berlangsung. Sebaliknya, jika Anda memulai operasi pengubahan ukuran saat instans mengelompokkan konfigurasi ulangnya.

Menggunakan penskalaan terkelola di Amazon EMR

Important

Kami sangat menyarankan Anda menggunakan rilis EMR Amazon terbaru (Amazon EMR 7.0.0) untuk penskalaan terkelola. Dalam beberapa rilis awal, Anda mungkin mengalami kegagalan aplikasi intermiten atau penundaan dalam penskalaan. Amazon EMR menyelesaikan masalah ini dengan rilis 5.x 5.30.2, 5.31.1, 5.32.1, 5.33.1 dan lebih

tinggi, dan dengan rilis 6.x 6.1.1, 6.2.1, 6.3.1 dan lebih tinggi. Untuk informasi selengkapnya Ketersediaan wilayah dan rilis, lihatKetersediaan penskalaan terkelola.

Gambaran Umum

Dengan Amazon EMR versi 5.30.0 dan yang lebih tinggi (kecuali Amazon EMR 6.0.0), Anda dapat mengaktifkan penskalaan terkelola Amazon EMR. Penskalaan terkelola memungkinkan Anda secara otomatis menambah atau mengurangi jumlah instance atau unit di klaster berdasarkan beban kerja. Amazon EMR terus mengevaluasi metrik klaster untuk membuat keputusan penskalaan yang mengoptimalkan kluster Anda untuk biaya dan kecepatan. Penskalaan terkelola tersedia untuk cluster yang terdiri dari grup instans atau armada instance.

Ketersediaan penskalaan terkelola

- Di Asia Pasifik (Jakarta), penskalaan terkelola Amazon EMR tersedia dengan Amazon EMR 6.14.0 dan lebih tinggi.
- Berikut iniWilayah AWS, penskalaan terkelola Amazon EMR tersedia dengan Amazon EMR 5.30.0 dan 6.1.0 dan yang lebih tinggi:

  AS Timur (Virginia N. dan Ohio), AS Barat (Oregon dan California N.), Amerika Selatan (Sao Paulo), Eropa (Frankfurt, Irlandia, London, Milan, Paris, dan Stockholm), Kanada (Tengah), Asia Pasifik (Hong Kong, Mumbai, Seoul, Singapura, Sydney, dan Tokyo), Timur Tengah (Bahrain), Afrika (Cape Town), (AS-Timur), (AS-Barat)), China (Beijing) dioperasikan oleh Sinnet, China AWS GovCloud (Ningxia) yang dioperasikan oleh NWCD. AWS GovCloud
- Penskalaan terkelola Amazon EMR hanya berfungsi dengan aplikasi YARN, seperti Spark, Hadoop, Hive, dan Flink. Itu tidak mendukung aplikasi yang tidak didasarkan pada YARN, seperti Presto dan HBase.

Parameter penskalaan terkelola

Anda harus mengonfigurasi parameter berikut untuk penskalaan terkelola. Batas hanya berlaku untuk simpul utama dan tugas. Anda tidak dapat menskalakan node utama setelah konfigurasi awal.

- Minimum (`MinimumCapacityUnits`) - Batas bawah kapasitas EC2 yang diizinkan dalam sebuah klaster. Hal ini diukur melalui inti atau instans virtual central processing unit (vCPU) untuk grup instans. Hal ini diukur melalui unit untuk armada instans.

- Maksimum (`MaximumCapacityUnits`) – Batas atas kapasitas EC2 yang diizinkan dalam sebuah klaster. Hal ini diukur melalui inti atau instans virtual central processing unit (vCPU) untuk grup instans. Hal ini diukur melalui unit untuk armada instans.
- Batas Sesuai Permintaan (`MaximumOnDemandCapacityUnits`) (Opsional) — Batas atas kapasitas EC2 yang diizinkan untuk jenis pasar Sesuai Permintaan dalam sebuah klaster. Jika parameter ini tidak ditentukan, default ke nilai `MaximumCapacityUnits`.
  - Parameter ini digunakan untuk memisah alokasi kapasitas antara Instans Sesuai Permintaan dan Instans Spot. Misalnya, jika Anda menetapkan parameter minimum sebagai 2 instans, parameter maksimum sebagai 100 instans, batas Sesuai Permintaan sebagai 10 instans, maka penskalaan terkelola Amazon EMR menskalakan hingga 10 Instans Sesuai Permintaan dan mengalokasikan kapasitas yang tersisa ke Instans Spot. Untuk informasi selengkapnya, lihat Skenario alokasi simpul.
- Simpul inti maksimum (`MaximumCoreCapacityUnits`) (Opsional) - Batas atas kapasitas EC2 yang diizinkan untuk tipe simpul inti dalam sebuah klaster. Jika parameter ini tidak ditentukan, default ke nilai `MaximumCapacityUnits`.
  - Parameter ini digunakan untuk memisah alokasi kapasitas antara simpul inti dan simpul tugas. Misalnya, jika Anda menetapkan parameter minimum sebagai 2 instance, maksimum 100 instance, node inti maksimum sebagai 17 instance, maka Amazon EMR mengelola skala skala hingga 17 node inti dan mengalokasikan 83 instance yang tersisa ke node tugas. Untuk informasi selengkapnya, lihat Skenario alokasi simpul.

Untuk informasi selengkapnya tentang parameter penskalaan terkelola, lihat `ComputeLimits`.

Pertimbangan untuk penskalaan terkelola Amazon EMR

- Penskalaan terkelola didukung dalam rilis EMR Amazon terbatas Wilayah AWS dan. Untuk informasi selengkapnya, lihat Ketersediaan penskalaan terkelola.
- Anda harus mengonfigurasi parameter yang diperlukan untuk penskalaan terkelola Amazon EMR. Untuk informasi selengkapnya, lihat Parameter penskalaan terkelola.
- Untuk menggunakan penskalaan terkelola, proses kolektor metrik harus dapat terhubung ke titik akhir API publik untuk penskalaan terkelola di API Gateway. Jika Anda menggunakan nama DNS pribadi denganAmazon Virtual Private Cloud, penskalaan terkelola tidak akan berfungsi dengan baik. Untuk memastikan bahwa penskalaan terkelola berfungsi, kami sarankan Anda melakukan salah satu tindakan berikut:
  - Hapus titik akhir VPC antarmuka API Gateway dari VPC Amazon Anda.

  - Ikuti petunjuk di Mengapa saya mendapatkan kesalahan HTTP 403 Forbidden saat menghubungkan ke API Gateway API saya dari VPC? untuk menonaktifkan pengaturan nama DNS pribadi.
  - Luncurkan cluster Anda di subnet pribadi sebagai gantinya. Untuk informasi lebih lanjut, lihat topik diSubnet privat.
- Jika pekerjaan YARN Anda sebentar-sebentar lambat selama penurunan skala, dan log Manajer Sumber Daya YARN menunjukkan bahwa sebagian besar node Anda dicantumkan penolakan selama waktu itu, Anda dapat menyesuaikan ambang batas waktu penonaktifan.

  Kurangi `spark.blacklist.decommissioning.timeout` dari satu jam menjadi satu menit untuk membuat node tersedia untuk wadah tertunda lainnya untuk melanjutkan pemrosesan tugas.

  Anda juga harus menyetel `YARN.resourcemanager.nodemanager-graceful-decommission-timeout-secs` ke nilai yang lebih besar untuk memastikan Amazon EMR tidak memaksa menghentikan node sementara “Spark Task” terpanjang masih berjalan di node. Default saat ini adalah 60 menit, yang berarti YARN menghentikan kontainer secara paksa setelah 60 menit setelah node memasuki status dekomisi.

  Contoh berikut baris YARN Resource Manager Log menunjukkan node yang ditambahkan ke status dekomisi:

```
2021-10-20 15:55:26,994 INFO
 org.apache.hadoop.YARN.server.resourcemanager.DefaultAMSProcessor
 (IPC Server handler 37 on default port 8030): blacklist are updated in
 Scheduler.blacklistAdditions: [ip-10-10-27-207.us-west-2.compute.internal,
 ip-10-10-29-216.us-west-2.compute.internal, ip-10-10-31-13.us-
west-2.compute.internal, ... , ip-10-10-30-77.us-west-2.compute.internal],
 blacklistRemovals: []
```

  Lihat detail selengkapnya tentang cara Amazon EMR terintegrasi dengan daftar penolakan YARN selama penonaktifan node, kasus ketika nodedi Amazon EMR dapat ditolak terdaftar, dan mengonfigurasi perilaku penonaktifan simpul Spark.
- Pemanfaatan volume EBS yang berlebihan dapat menyebabkan masalah Penskalaan Terkelola. Kami menyarankan Anda mempertahankan volume EBS di bawah 90% pemanfaatan. Untuk informasi selengkapnya, lihat Penyimpanan instans.
- CloudWatch Metrik Amazon sangat penting agar penskalaan terkelola Amazon EMR dapat beroperasi. Kami menyarankan Anda memantau CloudWatch metrik Amazon dengan cermat

untuk memastikan data tidak hilang. Untuk informasi selengkapnya tentang cara mengonfigurasi CloudWatch alarm untuk mendeteksi metrik yang hilang, lihat Menggunakan alarm Amazon CloudWatch .

- Operasi penskalaan terkelola pada klaster 5.30.0 dan 5.30.1 tanpa Presto yang diinstal dapat menyebabkan gagal aplikasi atau menyebabkan grup instans seragam atau armada instans tetap berada di negara `ARRESTED`, terutama ketika operasi menurunkan skala diikuti dengan cepat oleh operasi menaikkan skala.

  Sebagai solusinya, pilih Presto sebagai aplikasi untuk diinstal saat Anda membuat cluster dengan Amazon EMR rilis 5.30.0 dan 5.30.1, bahkan jika pekerjaan Anda tidak memerlukan Presto.
- Saat Anda menetapkan node inti maksimum dan batas On-Demand untuk penskalaan terkelola Amazon EMR, pertimbangkan perbedaan antara grup instans dan armada instans. Setiap grup instans terdiri dari tipe instans yang sama dan opsi pembelian yang sama untuk instans: Sesuai Permintaan atau Spot. Untuk setiap armada instans, Anda dapat menentukan hingga lima tipe instans, yang dapat disediakan sebagai instans Sesuai Permintaan dan instans Spot. Untuk informasi selengkapnya, lihat Membuat sebuah klaster dengan armada instans atau grup instans seragam, Opsi armada instans, dan Skenario alokasi simpul.
- Dengan Amazon EMR 5.30.0 dan yang lebih tinggi, jika Anda menghapus aturan default Izinkan Semua keluar ke 0.0.0.0/ untuk grup keamanan master, Anda harus menambahkan aturan yang memungkinkan konektivitas TCP keluar ke grup keamanan Anda untuk akses layanan pada port 9443. Grup keamanan Anda untuk akses layanan juga harus mengizinkan lalu lintas TCP masuk pada port 9443 dari grup keamanan utama. Untuk informasi selengkapnya tentang mengonfigurasi grup keamanan, lihat grup keamanan yang dikelola Amazon EMR untuk contoh utama (subnet pribadi).
- Penskalaan terkelola tidak mendukung fitur label node YARN. Hindari menggunakan label node pada cluster dengan penskalaan terkelola. Misalnya, jangan izinkan pelaksana berjalan hanya pada node tugas. Saat Anda menggunakan label node di kluster EMR Amazon, Anda mungkin menemukan bahwa klaster Anda tidak meningkat, yang dapat menyebabkan perlambatan aplikasi Anda.
- Anda dapat menggunakan AWS CloudFormation untuk mengonfigurasi penskalaan terkelola Amazon EMR. Untuk informasi selengkapnya, lihat AWS::EMR::Cluster dalam Panduan Pengguna AWS CloudFormation.

Riwayat fitur

Tabel ini mencantumkan pembaruan untuk kemampuan penskalaan terkelola Amazon EMR.

Tanggal rilis	Kemampuan	Amazon EMR versi
Oktober 10, 2023	Penskalaan terkelola tersedia di Wilayah `ap-southeast-3` Asia Pasifik (Jakarta).	6.14.0 dan lebih tinggi
Juli 28, 2023	Peningkatan penskalaan terkelola untuk beralih ke grup instans tugas yang berbeda saat Amazon EMR mengalami penundaan peningkatan skala dengan grup instans saat ini.	5.34.0 dan lebih tinggi, 6.4.0 dan lebih tinggi
Juni 16, 2023	Peningkatan penskalaan terkelola untuk mengetahui node yang menjalankan master aplikasi sehingga node tersebut tidak diperkecil. Untuk informasi selengkapnya, lihat Memahami strategi alokasi simpul dan skenario.	5.34.0 dan lebih tinggi, 6.4.0 dan lebih tinggi
Maret 21, 2022	Menambahkan kesadaran data shuffle Spark yang digunakan saat menskalakan cluster. Untuk klaster EMR Amazon dengan Apache Spark dan fitur penskalaan terkelola yang diaktifkan, Amazon EMR terus memantau pelaksana Spark dan lokasi data acak perantara. Dengan menggunakan informasi ini, Amazon EMR hanya mengurangi instance yang kurang dimanfaatkan yang tidak berisi data shuffle yang	5.34.0 dan lebih tinggi, 6.4.0 dan lebih tinggi

Tanggal rilis	Kemampuan	Amazon EMR versi
	digunakan secara aktif. Ini mencegah perhitungan ulang data shuffle yang hilang, membantu menurunkan biaya dan meningkatkan kinerja pekerjaan. Untuk informasi selengkapnya, lihat Panduan Pemrograman Spark.	

Mengkonfigurasi penskalaan terkelola untuk Amazon EMR

Bagian berikut menjelaskan cara meluncurkan cluster EMR yang menggunakan penskalaan terkelola denganAWS Management Console,, atau. AWS SDK for Java AWS Command Line Interface

Topik

- Gunakan AWS Management Console untuk mengkonfigurasi penskalaan terkelola
- Gunakan AWS CLI untuk mengkonfigurasi penskalaan terkelola
- Gunakan AWS SDK for Java untuk mengonfigurasi penskalaan terkelola

Gunakan AWS Management Console untuk mengkonfigurasi penskalaan terkelola

Anda dapat menggunakan konsol EMR Amazon untuk mengonfigurasi penskalaan terkelola saat membuat klaster atau mengubah kebijakan penskalaan terkelola untuk klaster yang sedang berjalan.

New console

Untuk mengonfigurasi penskalaan terkelola saat Anda membuat klaster dengan konsol baru

1. Masuk keAWS Management Console, dan buka konsol EMR Amazon di https://console.aws.amazon.com/emr.
2. Di bawah EMR pada EC2 di panel navigasi kiri, pilih Clusters, lalu pilih Create cluster.
3. Pilih rilis EMR Amazon emr-5.30.0 atau yang lebih baru, kecuali versi emr-6.0.0.
4. Di bawah opsi penskalaan dan penyediaan klaster, pilih Gunakan penskalaan yang dikelola EMR. Tentukan jumlah instans Minimum dan Maksimum, instans node inti Maksimum, dan instans Sesuai Permintaan Maksimum.

5. Pilih opsi lain yang berlaku untuk cluster Anda.
6. Untuk meluncurkan klaster Anda, pilih Buat klaster.

Untuk mengonfigurasi penskalaan terkelola pada cluster yang ada dengan konsol baru

1. Masuk keAWS Management Console, dan buka konsol EMR Amazon di https://console.aws.amazon.com/emr.
2. Di bawah EMR pada EC2 di panel navigasi kiri, pilih Cluster, dan pilih cluster yang ingin Anda perbarui.
3. Pada tab Instans pada halaman detail klaster, temukan bagian Pengaturan grup Instans. Pilih Edit penskalaan klaster untuk menentukan nilai baru untuk Jumlah instans Minimum dan Maksimum serta batas Sesuai Permintaan.

Old console

Saat membuat cluster di konsol lama, Anda dapat mengonfigurasi penskalaan terkelola menggunakan opsi cepat atau opsi konfigurasi klaster lanjutan. Anda juga dapat membuat atau mengubah kebijakan penskalaan terkelola untuk klaster yang sedang berjalan dengan memodifikasi pengaturan Penskalaan Terkelola pada halaman Ringkasan atau Perangkat Keras.

Untuk menggunakan opsi cepat untuk mengonfigurasi penskalaan terkelola saat Anda membuat klaster dengan konsol lama

1. Buka konsol Amazon EMR, pilih Buat klaster dan buka Buat Klaster - Opsi cepat.
2. Di bagian Konfigurasi perangkat keras di samping opsi penskalaan dan penyediaan klaster, pilih kotak centang untuk mengaktifkan node klaster skala berdasarkan beban kerja.
3. Di bawah Unit inti dan tugas, tentukan jumlah Minimum dan Maksimum instans inti dan instans tugas.

Untuk menggunakan opsi lanjutan untuk mengonfigurasi penskalaan terkelola saat Anda membuat klaster dengan konsol lama

1. Dalam konsol Amazon EMR, pilih Buat klaster, pilih Buka opsi lanjutan, pilih opsi untuk Langkah 1: Perangkat Lunak dan Langkah, dan kemudian buka Langkah 2: Konfigurasi perangkat keras.
2. Di bagian Komposisi klaster, pilih Armada Instans atau Grup instans seragam.

3. Di bawah opsi penskalaan dan penyediaan klaster, pilih Aktifkan penskalaan klaster. Kemudian pilih Gunakan skala terkelola EMR. Di bawah Unit inti dan tugas, tentukan jumlah Minimum dan Maksimum unit instans atau unit armada instans, Batas Sesuai Permintaan, dan Simpul Inti Maksimum.

   Untuk klaster yang terdiri dari grup instans, Anda juga dapat memilih Membuat kebijakan penskalaan otomatis kustom jika Anda ingin menentukan kebijakan penskalaan otomatis kustom untuk setiap kelompok instans. Untuk informasi selengkapnya, lihat Menggunakan penskalaan otomatis dengan kebijakan kustom untuk grup instans .

Untuk memodifikasi penskalaan terkelola pada cluster yang ada dengan konsol lama

1. Buka konsol Amazon EMR, pilih klaster Anda dari daftar klaster, dan kemudian pilih tab Perangkat keras.
2. Di bagian opsi penskalaan dan penyediaan klaster, pilih Edit untuk penskalaan terkelola EMR Amazon.
3. Di bagian opsi penskalaan dan penyediaan klaster, tentukan nilai baru untuk jumlah instans Minimum dan Maksimum serta batas On-Demand.

Gunakan AWS CLI untuk mengkonfigurasi penskalaan terkelola

Anda dapat menggunakan perintah AWS CLI untuk Amazon EMR untuk mengkonfigurasi penskalaan terkelola ketika Anda membuat sebuah klaster. Anda dapat menggunakan sintaks steno, menentukan konfigurasi JSON inline dalam perintah yang relevan, atau Anda dapat mereferensikan file yang berisi konfigurasi JSON. Anda juga dapat menerapkan kebijakan penskalaan terkelola ke klaster yang ada dan menghapus kebijakan penskalaan terkelola yang sebelumnya diterapkan. Selain itu, Anda dapat mengambil detail konfigurasi kebijakan penskalaan dari klaster berjalan.

Mengaktifkan Penskalaan Terkelola Selama Peluncuran Cluster

Anda dapat mengaktifkan penskalaan terkelola selama peluncuran klaster sebagaimana yang ditunjukkan oleh contoh berikut.

```
aws emr create-cluster \
 --service-role EMR_DefaultRole \
 --release-label emr-7.0.0 \
 --name EMR_Managed_Scaling_Enabled_Cluster \
 --applications Name=Spark Name=Hbase \
```

```
 --ec2-attributes KeyName=keyName,InstanceProfile=EMR_EC2_DefaultRole \
 --instance-groups InstanceType=m4.xlarge,InstanceGroupType=MASTER,InstanceCount=1
 InstanceType=m4.xlarge,InstanceGroupType=CORE,InstanceCount=2 \
 --region us-east-1 \
 --managed-scaling-policy
 ComputeLimits='{MinimumCapacityUnits=2,MaximumCapacityUnits=4,UnitType=Instances}'
```

Anda juga dapat menentukan konfigurasi kebijakan terkelola menggunakan managed-scaling-policy opsi -- saat Anda menggunakan`create-cluster`.

Menerapkan Kebijakan Penskalaan Terkelola ke Cluster yang Ada

Anda dapat menerapkan kebijakan penskalaan terkelola ke klaster yang ada sebagaimana yang ditunjukkan oleh contoh berikut.

```
aws emr put-managed-scaling-policy
--cluster-id j-123456
--managed-scaling-policy ComputeLimits='{MinimumCapacityUnits=1,
MaximumCapacityUnits=10,  MaximumOnDemandCapacityUnits=10, UnitType=Instances}'
```

Anda juga dapat menerapkan kebijakan penskalaan terkelola ke klaster yang sudah ada dengan menggunakan perintah `aws emr put-managed-scaling-policy`. Contoh berikut menggunakan referensi ke file JSON, `managedscaleconfig.json`, yang menentukan konfigurasi kebijakan penskalaan terkelola.

```
aws emr put-managed-scaling-policy --cluster-id j-123456 --managed-scaling-policy
 file://./managedscaleconfig.json
```

Contoh berikut menunjukkan isi file `managedscaleconfig.json`, yang mendefinisikan kebijakan penskalaan terkelola.

```
{
    "ComputeLimits": {
        "UnitType": "Instances",
        "MinimumCapacityUnits": 1,
        "MaximumCapacityUnits": 10,
        "MaximumOnDemandCapacityUnits": 10
    }
}
```

Mengambil Konfigurasi Kebijakan Penskalaan Terkelola

Perintah `GetManagedScalingPolicy` mengambil konfigurasi kebijakan. Sebagai contoh, perintah berikut ini mengambil konfigurasi untuk klaster dengan ID klaster `j-123456`.

```
aws emr get-managed-scaling-policy --cluster-id j-123456
```

Perintah tersebut menghasilkan output seperti berikut ini.

```
{
   "ManagedScalingPolicy": {
      "ComputeLimits": {
         "MinimumCapacityUnits": 1,
         "MaximumOnDemandCapacityUnits": 10,
         "MaximumCapacityUnits": 10,
         "UnitType": "Instances"
      }
   }
}
```

Untuk informasi selengkapnya tentang menggunakan perintah Amazon EMR dalam AWS CLI, lihat https://docs.aws.amazon.com/cli/latest/reference/emr.

Menghapus Kebijakan Penskalaan Terkelola

Perintah `RemoveManagedScalingPolicy` menghapus konfigurasi kebijakan. Sebagai contoh, perintah berikut menghapus konfigurasi untuk klaster dengan ID klaster `j-123456`.

```
aws emr remove-managed-scaling-policy --cluster-id j-123456
```

Gunakan AWS SDK for Java untuk mengonfigurasi penskalaan terkelola

Kutipan program berikut menunjukkan cara mengkonfigurasi penskalaan terkelola menggunakan AWS SDK for Java:

```
package com.amazonaws.emr.sample;

import java.util.ArrayList;
import java.util.List;

import com.amazonaws.AmazonClientException;
import com.amazonaws.auth.AWSCredentials;
import com.amazonaws.auth.AWSStaticCredentialsProvider;
import com.amazonaws.auth.profile.ProfileCredentialsProvider;
```

```
import com.amazonaws.regions.Regions;
import com.amazonaws.services.elasticmapreduce.AmazonElasticMapReduce;
import com.amazonaws.services.elasticmapreduce.AmazonElasticMapReduceClientBuilder;
import com.amazonaws.services.elasticmapreduce.model.Application;
import com.amazonaws.services.elasticmapreduce.model.ComputeLimits;
import com.amazonaws.services.elasticmapreduce.model.ComputeLimitsUnitType;
import com.amazonaws.services.elasticmapreduce.model.InstanceGroupConfig;
import com.amazonaws.services.elasticmapreduce.model.JobFlowInstancesConfig;
import com.amazonaws.services.elasticmapreduce.model.ManagedScalingPolicy;
import com.amazonaws.services.elasticmapreduce.model.RunJobFlowRequest;
import com.amazonaws.services.elasticmapreduce.model.RunJobFlowResult;

public class CreateClusterWithManagedScalingWithIG {

 public static void main(String[] args) {
  AWSCredentials credentialsFromProfile = getCreadentials("AWS-Profile-Name-Here");

  /**
   * Create an Amazon EMR client with the credentials and region specified in order to
 create the cluster
   */
  AmazonElasticMapReduce emr = AmazonElasticMapReduceClientBuilder.standard()
   .withCredentials(new AWSStaticCredentialsProvider(credentialsFromProfile))
   .withRegion(Regions.US_EAST_1)
   .build();

  /**
   * Create Instance Groups - Primary, Core, Task
   */
  InstanceGroupConfig instanceGroupConfigMaster = new InstanceGroupConfig()
    .withInstanceCount(1)
    .withInstanceRole("MASTER")
    .withInstanceType("m4.large")
    .withMarket("ON_DEMAND");

  InstanceGroupConfig instanceGroupConfigCore = new InstanceGroupConfig()
   .withInstanceCount(4)
   .withInstanceRole("CORE")
   .withInstanceType("m4.large")
   .withMarket("ON_DEMAND");

  InstanceGroupConfig instanceGroupConfigTask = new InstanceGroupConfig()
   .withInstanceCount(5)
   .withInstanceRole("TASK")
```

```
   .withInstanceType("m4.large")
   .withMarket("ON_DEMAND");

 List<InstanceGroupConfig> igConfigs = new ArrayList<>();
 igConfigs.add(instanceGroupConfigMaster);
 igConfigs.add(instanceGroupConfigCore);
 igConfigs.add(instanceGroupConfigTask);

       /**
        *  specify applications to be installed and configured when Amazon EMR creates
the cluster
        */
 Application hive = new Application().withName("Hive");
 Application spark = new Application().withName("Spark");
 Application ganglia = new Application().withName("Ganglia");
 Application zeppelin = new Application().withName("Zeppelin");

 /**
  * Managed Scaling Configuration -
        * Using UnitType=Instances for clusters composed of instance groups
  *
        * Other options are:
        * UnitType = VCPU ( for clusters composed of instance groups)
        * UnitType = InstanceFleetUnits ( for clusters composed of instance fleets)
        **/
 ComputeLimits computeLimits = new ComputeLimits()
   .withMinimumCapacityUnits(1)
   .withMaximumCapacityUnits(20)
   .withUnitType(ComputeLimitsUnitType.Instances);

 ManagedScalingPolicy managedScalingPolicy = new ManagedScalingPolicy();
 managedScalingPolicy.setComputeLimits(computeLimits);

 // create the cluster with a managed scaling policy
 RunJobFlowRequest request = new RunJobFlowRequest()
         .withName("EMR_Managed_Scaling_TestCluster")
         .withReleaseLabel("emr-7.0.0")          // Specifies the version label for
the Amazon EMR release; we recommend the latest release
         .withApplications(hive,spark,ganglia,zeppelin)
         .withLogUri("s3://path/to/my/emr/logs")  // A URI in S3 for log files is
required when debugging is enabled.
         .withServiceRole("EMR_DefaultRole")      // If you use a custom IAM service
role, replace the default role with the custom role.
```

```
          .withJobFlowRole("EMR_EC2_DefaultRole")  // If you use a custom Amazon EMR
 role for EC2 instance profile, replace the default role with the custom Amazon EMR
 role.
          .withInstances(new JobFlowInstancesConfig().withInstanceGroups(igConfigs)
              .withEc2SubnetId("subnet-123456789012345")
              .withEc2KeyName("my-ec2-key-name")
              .withKeepJobFlowAliveWhenNoSteps(true))
          .withManagedScalingPolicy(managedScalingPolicy);
    RunJobFlowResult result = emr.runJobFlow(request);

    System.out.println("The cluster ID is " + result.toString());
 }

 public static AWSCredentials getCredentials(String profileName) {
  // specifies any named profile in .aws/credentials as the credentials provider
  try {
   return new ProfileCredentialsProvider("AWS-Profile-Name-Here")
     .getCredentials();
        } catch (Exception e) {
            throw new AmazonClientException(
                    "Cannot load credentials from .aws/credentials file. " +
                    "Make sure that the credentials file exists and that the profile
 name is defined within it.",
                    e);
        }
 }

 public CreateClusterWithManagedScalingWithIG() { }
}
```

Memahami strategi alokasi simpul dan skenario

Bagian ini memberikan ikhtisar strategi alokasi node dan skenario penskalaan umum yang dapat Anda gunakan dengan penskalaan terkelola Amazon EMR.

Strategi alokasi simpul

Penskalaan terkelola Amazon EMR mengalokasikan node inti dan tugas berdasarkan strategi scale-up dan scale-down berikut:

Strategi penskalaan

- Penskalaan terkelola Amazon EMR pertama-tama menambahkan kapasitas ke node inti dan kemudian ke node tugas hingga kapasitas maksimum yang diizinkan tercapai atau hingga kapasitas target peningkatan skala yang diinginkan tercapai.
- Saat Amazon EMR mengalami penundaan peningkatan skala dengan grup instans saat ini, klaster yang menggunakan penskalaan terkelola secara otomatis beralih ke grup instans tugas yang berbeda.
- Jika parameter `MaximumCoreCapacityUnits` diatur, maka Amazon EMR menskalakan simpul inti sampai unit inti mencapai batas maksimum yang diizinkan. Semua kapasitas yang tersisa ditambahkan ke simpul tugas.
- Jika parameter `MaximumOnDemandCapacityUnits` diatur, maka Amazon EMR menskalakan klaster dengan menggunakan instans Sesuai Permintaan sampai unit Sesuai Permintaan mencapai batas maksimum yang diizinkan. Semua kapasitas yang tersisa ditambahkan menggunakan Instans Spot.
- Jika kedua parameter `MaximumCoreCapacityUnits` dan `MaximumOnDemandCapacityUnits` diatur, Amazon EMR mempertimbangkan kedua batas selama penskalaan.

  Misalnya, jika kurang dari`MaximumOnDemandCapacityUnits`, Amazon EMR pertama-tama menskalakan node inti hingga batas kapasitas inti tercapai. `MaximumCoreCapacityUnits` Untuk kapasitas yang tersisa, Amazon EMR pertama-tama menggunakan Instans Sesuai Permintaan untuk menskalakan node tugas hingga batas On-Demand tercapai, dan kemudian menggunakan Instans Spot untuk node tugas.

Strategi penskalaan

- Amazon EMR versi 5.34.0 dan yang lebih tinggi, dan Amazon EMR versi 6.4.0 dan lebih tinggi, mendukung penskalaan terkelola yang mengetahui data shuffle Spark (data yang didistribusikan ulang Spark di seluruh partisi untuk melakukan operasi tertentu). Untuk informasi selengkapnya tentang operasi shuffle, lihat Panduan Pemrograman Spark. Penskalaan penskalaan terkelola hanya instance yang kurang dimanfaatkan dan yang tidak berisi data shuffle yang digunakan secara aktif. Penskalaan cerdas ini mencegah hilangnya data shuffle yang tidak diinginkan, menghindari kebutuhan untuk upaya ulang pekerjaan dan perhitungan ulang data perantara.
- Penskalaan terkelola Amazon EMR pertama-tama menghapus node tugas dan kemudian menghapus node inti hingga kapasitas target penurunan skala yang diinginkan tercapai. Klaster tidak pernah menskalakan di bawah batas minimum dalam kebijakan penskalaan terkelola.
- Dalam setiap jenis node (baik node inti atau node tugas), penskalaan terkelola Amazon EMR menghapus Instans Spot terlebih dahulu dan kemudian menghapus Instans Sesuai Permintaan.

- Untuk cluster yang diluncurkan dengan Amazon EMR 5.x merilis 5.34.0 dan lebih tinggi, dan 6.x merilis 6.4.0 dan lebih tinggi, penskalaan Amazon EMR-managed tidak mengurangi node yang memiliki Apache Spark yang berjalan pada mereka. `ApplicationMaster` Ini meminimalkan kegagalan pekerjaan dan percobaan ulang, yang membantu meningkatkan kinerja pekerjaan dan mengurangi biaya. Untuk mengonfirmasi node mana di cluster Anda yang sedang berjalan`ApplicationMaster`, kunjungi Spark History Server dan filter driver di bawah tab Executors ID aplikasi Spark Anda.

Jika klaster tidak memiliki beban apapun, maka Amazon EMR membatalkan penambahan instans baru dari evaluasi sebelumnya dan melakukan operasi menurunkan skala. Jika klaster memiliki beban berat, Amazon EMR membatalkan penghapusan instans dan melakukan operasi menaikkan skala.

Pertimbangan alokasi simpul

Kami merekomendasikan Anda menggunakan opsi pembelian Sesuai Permintaan untuk simpul inti untuk menghindari kehilangan data HDFS dalam kasus reklamasi Spot. Anda dapat menggunakan opsi pembelian Spot untuk simpul tugas untuk mengurangi biaya dan mendapatkan eksekusi pekerjaan yang lebih cepat ketika lebih banyak Instans Spot ditambahkan ke simpul tugas.

Skenario alokasi simpul

Anda dapat membuat berbagai skenario penskalaan berdasarkan kebutuhan Anda dengan mengatur parameter maksimum, minimum, batas Sesuai Permintaan, dan maksimum simpul inti dalam kombinasi yang berbeda.

Skenario 1: Saja Skala Node Inti

Untuk menskalakan simpul inti saja, parameter penskalaan terkelola harus memenuhi persyaratan berikut:

- Batas Sesuai Permintaan sama dengan batas maksimum.
- Simpul inti maksimum sama dengan batas maksimum.

Ketika batas Sesuai Permintaan dan parameter simpul inti maksimum tidak ditentukan, kedua parameter default ke batas maksimum.

Contoh berikut menunjukkan skenario penskalaan simpul inti saja.

Status awal klaster	Parameter penskalaan	Perilaku penskalaan
Grup instans Inti: 1 Sesuai Permintaan Tugas: 1 Sesuai Permintaan dan 1 Spot	`UnitType`: Instans `MinimumCapacityUnits` : 1 `MaximumCapacityUnits` : 20 `MaximumOnDemandCapacityUnits` : 20 `MaximumCoreCapacityUnits` : 20	Menskalakan antara 1 hingga 20 Instans atau unit armada instans pada simpul inti menggunakan jenis Sesuai Permintaan. Tidak ada penskalaan pada simpul tugas.
Armada contoh Inti: 1 Sesuai Permintaan Tugas: 1 Sesuai Permintaan dan 1 Spot	UnitType: InstanceFleetUnits `MinimumCapacityUnits` : 1 `MaximumCapacityUnits` : 20 `MaximumOnDemandCapacityUnits` : 20 `MaximumCoreCapacityUnits` : 20	

Skenario 2: Skalakan node tugas saja

Untuk menskalakan simpul tugas saja, parameter penskalaan terkelola harus memenuhi persyaratan berikut:

- Simpul inti maksimum harus sama dengan batas minimum.

Contoh berikut menunjukkan skenario penskalaan simpul tugas saja.

Status awal klaster	Parameter penskalaan	Perilaku penskalaan

Status awal klaster	Parameter penskalaan	Perilaku penskalaan
Grup instans Inti: 2 Sesuai Permintaan Tugas: 1 Spot	`UnitType`: Instans `MinimumCapacityUnits` : 2 `MaximumCapacityUnits` : 20 `MaximumCoreCapacityUnits` : 2	Jaga agar simpul inti tetap stabil pada 2 dan hanya menskalakan simpul tugas antara 0 hingga 18 instans atau unit armada instans. Kapasitas antara batas minimum dan maksimum ditambahkan ke simpul tugas saja.
Armada contoh Inti: 2 Sesuai Permintaan Tugas: 1 Spot	`UnitType`: InstanceFleetUnits `MinimumCapacityUnits` : 2 `MaximumCapacityUnits` : 20 `MaximumCoreCapacityUnits` : 2	

Skenario 3: Hanya Instans Sesuai Permintaan di klaster

Untuk memiliki Instans Sesuai Permintaan saja, klaster Anda dan parameter penskalaan terkelola harus memenuhi persyaratan berikut:

- Batas Sesuai Permintaan sama dengan batas maksimum.

  Ketika batas Sesuai Permintaan tidak ditentukan, nilai parameter default ke batas maksimum. Nilai default menunjukkan bahwa Amazon EMR menskalakan Instans Sesuai Permintaan saja.

Jika simpul inti maksimum kurang dari batas maksimum, parameter simpul inti maksimum dapat digunakan untuk membagi alokasi kapasitas antara simpul inti dan tugas.

Untuk mengaktifkan skenario ini dalam sebuah klaster yang terdiri dari grup instans, semua kelompok simpul dalam klaster harus menggunakan tipe pasar Sesuai Permintaan selama konfigurasi awal.

Contoh berikut menunjukkan skenario dari memiliki Instans Sesuai Permintaan di seluruh klaster.

Status awal klaster	Parameter penskalaan	Perilaku penskalaan
Grup instans Inti: 1 Sesuai Permintaan Tugas: 1 Sesuai Permintaan	`UnitType`: Instans `MinimumCapacityUnits` : 1 `MaximumCapacityUnits` : 20 `MaximumOnDemandCapacityUnits` : 20 `MaximumCoreCapacityUnits` : 12	Menskalakan antara 1 hingga 12 Instans atau unit armada instans pada simpul inti menggunakan tipe Sesuai Permintaan. Menskalakan kapasitas yang tersisa menggunakan Sesuai Permintaan pada simpul tugas. Tidak ada penskalaan menggunakan Instans Spot.
Armada contoh Inti: 1 Sesuai Permintaan Tugas: 1 Sesuai Permintaan	`UnitType`: InstanceFleetUnits `MinimumCapacityUnits` : 1 `MaximumCapacityUnits` : 20 `MaximumOnDemandCapacityUnits` : 20 `MaximumCoreCapacityUnits` : 12	

Skenario 4: Hanya Instance Spot di cluster

Untuk memiliki Instans Spot saja, klaster Anda dan parameter penskalaan terkelola harus memenuhi persyaratan berikut:

- Batas Sesuai Permintaan diatur ke 0.

Jika simpul inti maksimum kurang dari batas maksimum, parameter simpul inti maksimum dapat digunakan untuk membagi alokasi kapasitas antara simpul inti dan tugas.

Untuk mengaktifkan skenario ini dalam sebuah klaster yang terdiri dari grup instans, grup instans inti harus menggunakan opsi pembelian Spot selama konfigurasi awal. Jika tidak ada Instans Spot di

grup instance tugas, penskalaan terkelola Amazon EMR akan membuat grup tugas menggunakan Instans Spot bila diperlukan.

Contoh berikut menunjukkan skenario dari memiliki Instans Spot di seluruh klaster.

Status awal klaster	Parameter penskalaan	Perilaku penskalaan
Grup instans Inti: 1 Spot Tugas: 1 Spot	`UnitType`: Instans `MinimumCapacityUnits` : 1 `MaximumCapacityUnits` : 20 `MaximumOnDemandCapacityUnits` : 0	Menskalakan antara 1 hingga 20 Instans atau unit armada instans pada simpul inti menggunakan Spot. Tidak ada penskalaan menggunakan tipe Sesuai Permintaan.
Armada contoh Inti: 1 Spot Tugas: 1 Spot	`UnitType`: InstanceFleetUnits `MinimumCapacityUnits` : 1 `MaximumCapacityUnits` : 20 `MaximumOnDemandCapacityUnits` : 0	

Skenario 5: Skala Instans Sesuai Permintaan pada node inti dan Instans Spot pada node tugas

Untuk menskalakan Instans Sesuai Permintaan pada simpul inti dan Instans Spot pada simpul tugas, parameter penskalaan terkelola harus memenuhi persyaratan berikut:

- Batas Sesuai Permintaan harus sama dengan simpul inti maksimum.
- Batas Sesuai Permintaan dan simpul inti maksimum harus kurang dari batas maksimum.

Untuk mengaktifkan skenario ini dalam sebuah klaster yang terdiri dari grup instans, grup simpul inti harus menggunakan opsi pembelian Sesuai Permintaan.

Contoh berikut menunjukkan skenario penskalaan Instans Sesuai Permintaan pada simpul inti dan Instans Spot pada simpul tugas.

Status awal klaster	Parameter penskalaan	Perilaku penskalaan
Grup instans Inti: 1 Sesuai Permintaan Tugas: 1 Sesuai Permintaan dan 1 Spot	`UnitType`: Instans `MinimumCapacityUnits` : 1 `MaximumCapacityUnits` : 20 `MaximumOnDemandCapacityUnits` : 7 `MaximumCoreCapacityUnits` : 7	Menaikkan skala hingga 6 unit Sesuai Permintaan pada simpul inti karena sudah ada 1 unit Sesuai Permintaan pada simpul tugas dan batas maksimum untuk Sesuai Permintaan adalah 7. Kemudian naikkan skala hingga 13 unit Spot pada simpul tugas.
Armada contoh Inti: 1 Sesuai Permintaan Tugas: 1 Sesuai Permintaan dan 1 Spot	`UnitType`: InstanceFleetUnits `MinimumCapacityUnits` : 1 `MaximumCapacityUnits` : 20 `MaximumOnDemandCapacityUnits` : 7 `MaximumCoreCapacityUnits` : 7	

Memahami metrik penskalaan terkelola

Amazon EMR menerbitkan metrik resolusi tinggi dengan data pada rincian satu menit ketika penskalaan terkelola diaktifkan untuk suatu klaster. Anda dapat melihat peristiwa pada setiap inisiasi dan penyelesaian pengubahan ukuran yang dikendalikan oleh penskalaan terkelola dengan konsol EMR Amazon atau konsol Amazon. CloudWatch CloudWatch metrik sangat penting agar penskalaan terkelola Amazon EMR dapat beroperasi. Kami menyarankan Anda memantau CloudWatch metrik dengan cermat untuk memastikan data tidak hilang. Untuk informasi selengkapnya tentang cara mengonfigurasi CloudWatch alarm untuk mendeteksi metrik yang hilang, lihat Menggunakan alarm

Amazon CloudWatch . Untuk informasi selengkapnya tentang penggunaan CloudWatch peristiwa dengan Amazon EMR, lihat Memantau CloudWatch peristiwa.

Metrik berikut menunjukkan kapasitas saat ini atau kapasitas target suatu klaster. Metrik ini hanya tersedia apabila penskalaan terkelola diaktifkan. Untuk klaster yang terdiri dari armada instans, metrik kapasitas klaster diukur dalam `Units`. Untuk klaster yang terdiri dari grup instans, metrik kapasitas klaster diukur dalam `Nodes` atau `vCPU` berdasarkan jenis unit yang digunakan dalam kebijakan penskalaan terkelola.

Metrik	Deskripsi
• `TotalUnitsRequested` • `TotalNodesRequested` • `TotalVCPURequested`	Target jumlah total unit/simpul/vCPU dalam sebuah klaster yang ditentukan oleh penskalaan terkelola. Unit: Jumlah
• `TotalUnitsRunning` • `TotalNodesRunning` • `TotalVCPURunning`	Jumlah total unit/simpul/vCPU saat ini yang tersedia dalam klaster yang sedang berjalan. Ketika ada permintaan perubahan ukuran klaster, metrik ini akan diperbarui setelah instans baru ditambahkan atau dihapus dari klaster. Unit: Jumlah
• `CoreUnitsRequested` • `CoreNodesRequested` • `CoreVCPURequested`	Target jumlah unit/simpul/vCPU INTI dalam sebuah klaster yang ditentukan oleh penskalaan terkelola. Unit: Jumlah
• `CoreUnitsRunning` • `CoreNodesRunning` •	Jumlah unit/simpul/vCPU INTI saat ini yang berjalan dalam suatu klaster. Unit: Jumlah

Metrik	Deskripsi
`CoreVCPURunning`	
• `TaskUnitsRequested` • `TaskNodesRequested` • `TaskVCPURequested`	Jumlah target unit/simpul/vCPU TUGAS dalam sebuah klaster yang ditentukan oleh penskalaan terkelola. Unit: Jumlah
• `TaskUnitsRunning` • `TaskNodesRunning` • `TaskVCPURunning`	Jumlah unit/simpul/vCPU TUGAS saat ini yang berjalan dalam suatu klaster. Unit: Jumlah

Metrik berikut menunjukkan status penggunaan klaster dan aplikasi. Metrik ini tersedia untuk semua fitur Amazon EMR, tetapi diterbitkan pada resolusi yang lebih tinggi dengan data pada rincian satu menit ketika penskalaan terkelola diaktifkan untuk sebuah klaster. Anda dapat mengkorelasikan metrik berikut dengan metrik kapasitas klaster di tabel sebelumnya untuk memahami keputusan penskalaan terkelola.

Metrik	Deskripsi
`AppsCompleted`	Jumlah aplikasi yang dikirimkan ke YARN yang telah selesai. Kasus penggunaan: Memantau kemajuan klaster Unit: Jumlah
`AppsPending`	Jumlah aplikasi yang dikirimkan ke YARN yang berada dalam status tertunda.

Metrik	Deskripsi
	Kasus penggunaan: Memantau kemajuan klaster Unit: Jumlah
AppsRunning	Jumlah aplikasi yang dikirimkan ke YARN yang sedang berjalan. Kasus penggunaan: Memantau kemajuan klaster Unit: Jumlah
ContainerAllocated	Jumlah wadah sumber daya yang dialokasikan oleh. ResourceManager Kasus penggunaan: Memantau kemajuan klaster Unit: Jumlah
ContainerPending	Jumlah kontainer dalam antrean yang belum dialokasikan. Kasus penggunaan: Memantau kemajuan klaster Unit: Jumlah
ContainerPendingRatio	Rasio kontainer yang tertunda dengan kontainer yang dialokasikan (ContainerPendingRatio = ContainerPending / ContainerAllocated). Jika ContainerAllocated = 0, maka ContainerPendingRatio =ContainerPending. Nilai Container PendingRatio mewakili angka, bukan persentase. Nilai ini berguna untuk menskalakan sumber daya klaster berdasark an perilaku alokasi kontainer. Unit: Jumlah

Metrik	Deskripsi
`HDFSUtilization`	Persentase penyimpanan HDFS yang saat ini digunakan. Kasus penggunaan: Menganalisis performa klaster Unit: Persen
`IsIdle`	Menunjukkan bahwa klaster tidak lagi melakukan pekerjaan , tetapi masih hidup dan menimbulkan biaya. Diatur ke 1 jika tidak ada tugas yang berjalan dan tidak ada pekerjaan yang berjalan, dan diatur ke 0 jika sebaliknya. Nilai ini diperiksa pada interval lima menit dan nilai 1 hanya menunjukkan bahwa klaster tersebut menganggur ketika diperiksa, bukan bahwa klaster tersebut menganggur selama lima menit tersebut. Untuk menghindari positif yang salah, Anda harus menyalakan alarm ketika nilai ini 1 selama lebih dari satu pemeriksaan lima menit berturut-turut. Misalnya, Anda mungkin menyalakan alarm pada nilai ini jika telah 1 selama tiga puluh menit atau lebih. Kasus penggunaan: Memantau performa klaster Unit: Boolean
`MemoryAvailableMB`	Jumlah memori yang tersedia untuk dialokasikan. Kasus penggunaan: Memantau kemajuan klaster Unit: Jumlah

Metrik	Deskripsi
`MRActiveNodes`	Jumlah node yang saat ini menjalankan MapReduce tugas atau pekerjaan. Setara dengan metrik YARN `mapred.resourcemanager.NoOfActiveNodes` . Kasus penggunaan: Memantau kemajuan klaster Unit: Jumlah
`YARNMemoryAvailablePercentage`	Persentase sisa memori yang tersedia untuk YARN (YARN MemoryAvailablePercentage = MemoryAvailable MB/ MemoryTotalMB). Nilai ini berguna untuk menskalakan sumber daya klaster berdasarkan penggunaan memori YARN. Unit: Persen

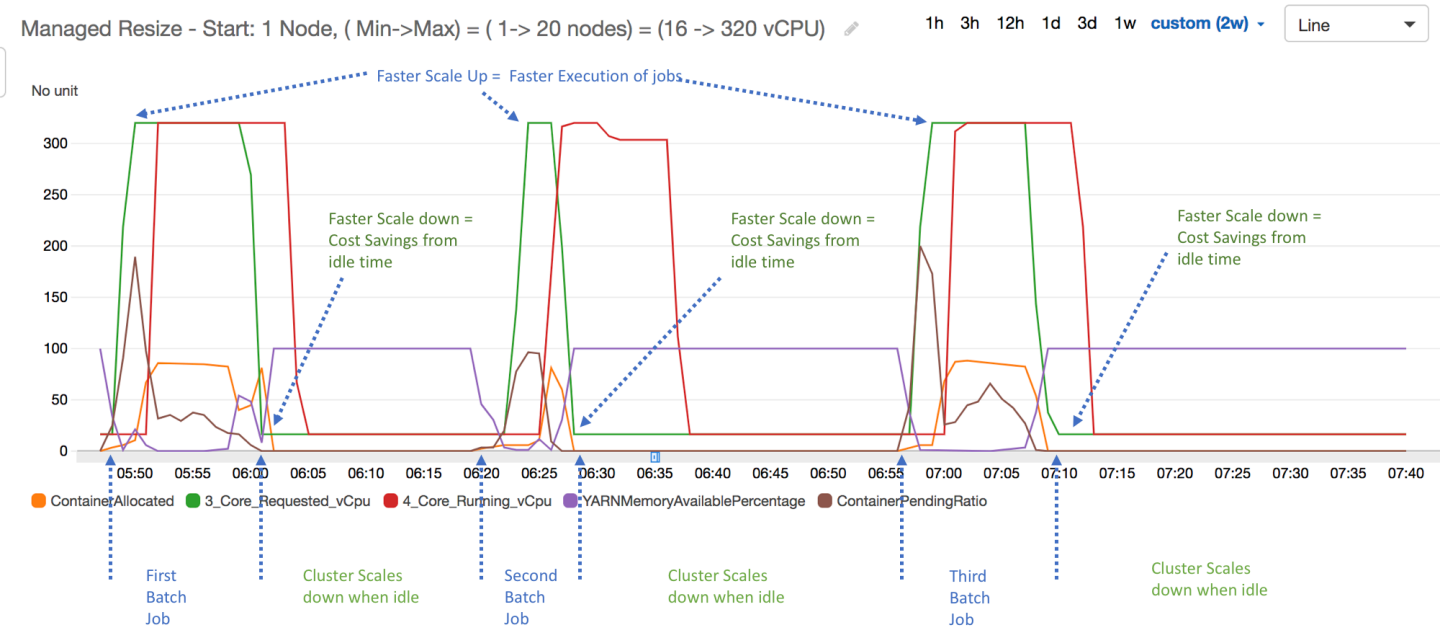
Membuat grafik metrik penskalaan terkelola

Anda dapat membuat grafik metrik untuk memvisualisasikan pola beban kerja klaster Anda dan keputusan penskalaan terkait yang dibuat oleh penskalaan terkelola Amazon EMR seperti yang ditunjukkan oleh langkah-langkah berikut.

Untuk membuat grafik metrik penskalaan terkelola di konsol CloudWatch

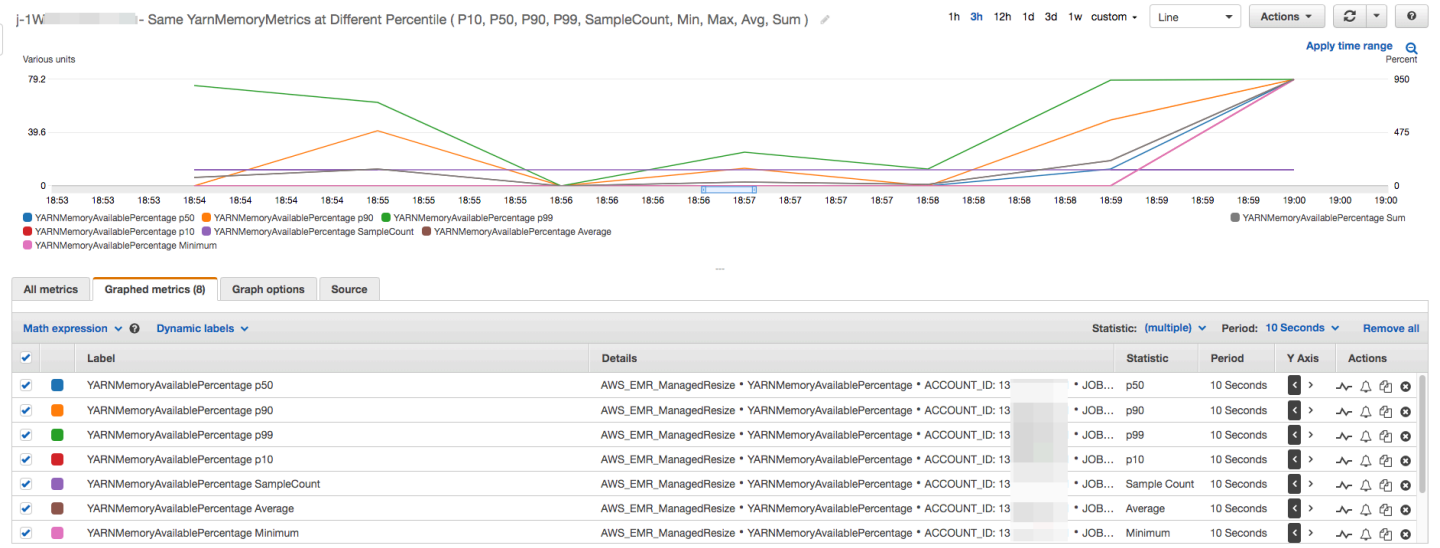
1. Buka konsol CloudWatch.
2. Di panel navigasi, pilih Amazon EMR. Anda dapat mencari di pengidentifikasi klaster pada klaster tersebut untuk memantau.
3. Gulir ke bawah ke metrik untuk membuat grafik. Buka metrik untuk menampilkan grafik.
4. Untuk membuat grafik pada satu metrik atau lebih, pilih kotak centang di samping setiap metrik.

Contoh berikut menggambarkan aktivitas penskalaan terkelola Amazon EMR dari sebuah cluster. Grafik menunjukkan tiga periode penskalaan otomatis, yang menghemat biaya ketika ada beban kerja yang kurang aktif.



Semua metrik penggunaan dan kapasitas klaster dipublikasikan pada interval satu menit. Informasi statistik tambahan juga dikaitkan dengan setiap data satu menit, yang memungkinkan Anda merencanakan berbagai fungsi seperti `Percentiles`, `Min`, `Max`, `Sum`, `Average`, `SampleCount`.

Misalnya, grafik berikut menggambarkan metrik `YARNMemoryAvailablePercentage` yang sama pada persentil yang berbeda, P10, P50, P90, P99, bersama dengan `Sum`, `Average`, `Min`, `SampleCount`.



Menggunakan penskalaan otomatis dengan kebijakan kustom untuk grup instans

Penskalaan otomatis dengan kebijakan khusus di Amazon EMR rilis 4.0 dan yang lebih tinggi memungkinkan Anda untuk menskalakan dan menskalakan secara terprogram di node inti dan node tugas berdasarkan metrik dan parameter lain CloudWatch yang Anda tentukan dalam kebijakan penskalaan. Penskalaan otomatis dengan kebijakan kustom tersedia dengan konfigurasi grup instans dan tidak tersedia jika Anda menggunakan armada instans. Untuk informasi selengkapnya tentang grup instans dan armada instans, lihat Membuat sebuah klaster dengan armada instan atau grup instans seragam.

> **Note**
>
> Untuk menggunakan penskalaan otomatis dengan fitur kebijakan kustom di Amazon EMR, Anda harus mengatur `true` untuk parameter `VisibleToAllUsers` saat Anda membuat sebuah klaster. Untuk informasi lebih lanjut, lihat SetVisibleToAllUsers.

Kebijakan penskalaan adalah bagian dari konfigurasi grup instans. Anda dapat menentukan kebijakan selama konfigurasi awal grup instans, atau dengan memodifikasi grup instans di klaster yang ada, bahkan ketika grup instans tersebut aktif. Setiap grup instans dalam klaster, kecuali grup instans utama, dapat memiliki kebijakan penskalaannya sendiri, yang terdiri dari aturan scale-out dan scale-in. Aturan penskalaan keluar dan penskalaan ke dalam dapat dikonfigurasi secara independen, dengan parameter yang berbeda untuk setiap aturan.

Anda dapat mengonfigurasi kebijakan penskalaan dengan AWS Management ConsoleAWS CLI, API EMR Amazon, atau Amazon. Saat Anda menggunakan AWS CLI atau API Amazon EMR, Anda menentukan kebijakan penskalaan dalam format JSON. Selain itu, saat menggunakan AWS CLI atau Amazon EMR API, Anda dapat menentukan metrik kustom CloudWatch . Metrik khusus tidak tersedia untuk dipilih dengan. AWS Management Console Saat Anda pertama kali membuat kebijakan penskalaan dengan konsol, kebijakan default yang cocok untuk banyak aplikasi sudah dikonfigurasi sebelumnya untuk membantu Anda memulai. Anda dapat menghapus atau mengubah aturan default.

Meskipun penskalaan otomatis memungkinkan Anda menyesuaikan on-the-fly kapasitas klaster EMR, Anda tetap harus mempertimbangkan persyaratan beban kerja dasar dan merencanakan konfigurasi grup node dan instans Anda. Untuk informasi selengkapnya, lihat Panduan konfigurasi klaster.

> **Note**
>
> Untuk sebagian besar beban kerja, disarankan untuk mengatur aturan penskalaan ke dalam dan penskalaan keluar untuk mengoptimalkan pemanfaatan sumber daya. Mengatur baik aturan tanpa cara lain yang Anda butuhkan untuk secara manual mengubah ukuran jumlah instans setelah aktivitas penskalaan. Dengan kata lain, hal ini mengatur kebijakan penskalaan keluar atau ke dalam otomatis "satu arah" dengan pengaturan ulang manual.

Membuat IAM role untuk penskalaan otomatis

Penskalaan otomatis di Amazon EMR memerlukan IAM role dengan izin untuk menambahkan dan mengakhiri instans saat aktivitas penskalaan terpicu. Peran default yang dikonfigurasi dengan kebijakan peran dan kebijakan kepercayaan yang sesuai, `EMR_AutoScaling_DefaultRole`, tersedia untuk tujuan ini. Saat Anda membuat klaster dengan kebijakan penskalaan untuk pertama kalinya denganAWS Management Console, Amazon EMR akan membuat peran default dan melampirkan kebijakan terkelola default untuk izin,. `AmazonElasticMapReduceforAutoScalingRole`

Bila Anda membuat klaster dengan kebijakan penskalaan otomatis denganAWS CLI, Anda harus terlebih dahulu memastikan bahwa peran IAM default ada, atau bahwa Anda memiliki peran IAM kustom dengan kebijakan terlampir yang menyediakan izin yang sesuai. Untuk membuat peran default, Anda dapat menjalankan perintah `create-default-roles` sebelum Anda membuat sebuah klaster. Anda kemudian dapat menentukan opsi `--auto-scaling-role EMR_AutoScaling_DefaultRole` saat Anda membuat sebuah klaster. Atau, Anda dapat membuat peran penskalaan otomatis kustom dan kemudian menentukannya ketika Anda membuat sebuah klaster, misalnya `--auto-scaling-role` *`MyEMRAutoScalingRole`*. Jika Anda membuat peran penskalaan otomatis disesuaikan untuk Amazon EMR, sebaiknya Anda mendasarkan kebijakan izin untuk peran kustom Anda berdasarkan kebijakan terkelola. Untuk informasi selengkapnya, lihat Konfigurasi peran layanan IAM untuk izin Amazon EMR untuk layanan AWS dan sumber daya.

Memahami aturan penskalaan otomatis

Ketika aturan penskalaan keluar memicu aktivitas penskalaan untuk grup instans, instans Amazon EC2 ditambahkan ke grup instans sesuai dengan aturan Anda. Simpul baru dapat digunakan oleh aplikasi seperti Apache Spark, Apache Hive, dan Presto segera setelah instans Amazon EC2 memasuki status `InService`. Anda juga dapat membuat aturan penskalaan ke dalam yang mengakhiri instans dan menghapus instans. Untuk informasi selengkapnya tentang siklus hidup

instans Amazon EC2 yang diskalakan secara otomatis, lihat siklus hidup Auto Scaling dalam Panduan Pengguna Amazon EC2 Auto Scaling.

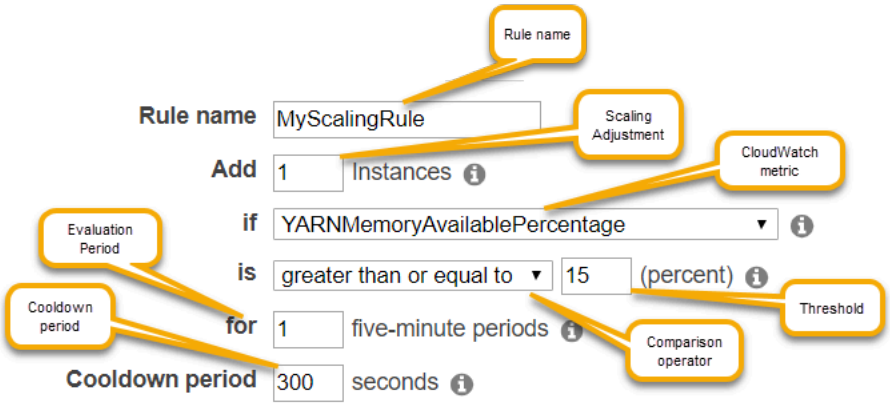
Anda dapat mengonfigurasi cara klaster mengakhiri instans Amazon EC2. Anda dapat memilih untuk mengakhiri pada batasan jam instans Amazon EC2 untuk penagihan, atau saat tugas selesai. Pengaturan ini berlaku baik untuk penskalaan otomatis dan operasi perubahan ukuran manual. Untuk informasi selengkapnya tentang konfigurasi ini, lihat Menurunkan skala klaster.

Parameter berikut ini untuk setiap aturan dalam kebijakan menentukan perilaku penskalaan otomatis.

> **Note**
>
> Parameter yang tercantum di sini didasarkan pada AWS Management Console untuk Amazon EMR. Saat Anda menggunakan AWS CLI atau API Amazon EMR, tersedia opsi konfigurasi lanjutan tambahan. Untuk informasi selengkapnya tentang opsi lanjutan, lihat SimpleScalingPolicyConfigurationdi Referensi API EMR Amazon.

- Instans maksimum dan instans minimum. Batasan Instans maksimum menentukan jumlah maksimum instans Amazon EC2 yang dapat berada dalam grup instans, dan berlaku untuk semua aturan penskalaan keluar. Demikian pula, batasan Instans Minimum menentukan jumlah minimum instans Amazon EC2 dan berlaku untuk semua aturan penskalaan ke dalam.
- Nama Aturan, yang harus unik dalam kebijakan.
- Penyesuaian penskalaan, yang menentukan jumlah instans EC2 untuk ditambahkan (untuk aturan penskalaan keluar) atau diakhiri (untuk aturan penskalaan ke dalam) selama aktivitas penskalaan dipicu oleh aturan.
- CloudWatch Metrik, yang diawasi untuk kondisi alarm.
- Operator perbandingan, yang digunakan untuk membandingkan CloudWatch metrik dengan nilai Threshold dan menentukan kondisi pemicu.
- Periode evaluasi, dalam peningkatan lima menit, di mana CloudWatch metrik harus dalam kondisi pemicu sebelum aktivitas penskalaan dipicu.
- Periode pendinginan, dalam detik, yang menentukan jumlah waktu yang harus berlalu antara aktivitas penskalaan yang dimulai oleh aturan dan dimulainya aktivitas penskalaan berikutnya, terlepas dari aturan yang memicunya. Ketika grup instans telah menyelesaikan aktivitas penskalaan dan mencapai status pasca-skala, periode cooldown memberikan kesempatan bagi CloudWatch metrik yang mungkin memicu aktivitas penskalaan berikutnya untuk stabil. Untuk

informasi selengkapnya, lihat pendinginan Auto Scaling dalam Panduan Pengguna Amazon EC2 Auto Scaling.



Pertimbangan dan batasan

- CloudWatch Metrik Amazon sangat penting agar penskalaan otomatis Amazon EMR dapat beroperasi. Kami menyarankan Anda memantau CloudWatch metrik Amazon dengan cermat untuk memastikan data tidak hilang. Untuk informasi selengkapnya tentang cara mengonfigurasi CloudWatch alarm Amazon untuk mendeteksi metrik yang hilang, lihat Menggunakan alarm Amazon CloudWatch .
- Pemanfaatan volume EBS yang berlebihan dapat menyebabkan masalah Penskalaan Terkelola. Kami menyarankan Anda memantau penggunaan volume EBS dengan cermat untuk memastikan volume EBS di bawah 90% pemanfaatan. Lihat Penyimpanan instans untuk informasi tentang menentukan volume EBS tambahan.
- Penskalaan otomatis dengan kebijakan khusus di Amazon EMR rilis 5.18 hingga 5.28 mungkin mengalami kegagalan penskalaan yang disebabkan oleh data yang sebentar-sebentar hilang dalam metrik Amazon. CloudWatch Kami menyarankan Anda menggunakan versi EMR Amazon terbaru untuk penskalaan otomatis yang lebih baik. Anda juga dapat menghubungi AWSSupport untuk patch jika Anda perlu menggunakan rilis Amazon EMR antara 5.18 dan 5.28.

Menggunakan AWS Management Console untuk mengonfigurasi penskalaan otomatis

Saat membuat klaster, Anda mengonfigurasi kebijakan penskalaan untuk grup instans dengan opsi konfigurasi klaster lanjutan. Anda juga dapat membuat atau mengubah kebijakan penskalaan untuk grup instans dalam layanan dengan memodifikasi grup instans di pengaturan Perangkat keras klaster yang ada.

> **Note**
>
> Konsol EMR Amazon baru (https://console.aws.amazon.com/emr) menggunakan penskalaan terkelola alih-alih penskalaan otomatis. Untuk menggunakan penskalaan otomatis, pastikan Anda masuk ke konsol lama di https://console.aws.amazon.com/elasticmapreduce.

1. Arahkan ke konsol EMR Amazon baru dan pilih Beralih ke konsol lama dari navigasi samping. Untuk informasi selengkapnya tentang apa yang diharapkan saat beralih ke konsol lama, lihat Menggunakan konsol lama.
2. Jika Anda membuat sebuah klaster, di konsol Amazon EMR, pilih Buat Klaster, pilih Buka opsi lanjutan, pilih opsi untuk Langkah 1: Perangkat Lunak dan Langkah, dan kemudian buka Langkah 2: Konfigurasi Perangkat Keras.

   - atau -

   Jika Anda memodifikasi grup instans di klaster berjalan, pilih klaster Anda dari daftar klaster, dan kemudian perluas bagian Perangkat keras.
3. Di bagian opsi penskalaan dan penyediaan kluster, pilih Aktifkan penskalaan klaster. Kemudian pilih Membuat kebijakan penskalaan otomatis kustom.

   Dalam tabel Kebijakan penskalaan otomatis kustom, klik ikon pensil yang muncul di baris grup instans yang ingin Anda konfigurasikan. Layar Aturan Auto Scaling terbuka.
4. Ketik Instans maksimum yang Anda inginkan untuk berada dalam grup instans setelah penskalaan keluar, dan ketik Instans Minimum yang Anda inginkan untuk berada dalam grup instans setelah penskalaan ke dalam.
5. Klik pensil untuk mengedit parameter aturan, klik X untuk menghapus aturan dari kebijakan, dan klik Tambahkan Aturan untuk menambahkan aturan tambahan.
6. Pilih parameter aturan seperti yang dijelaskan sebelumnya dalam topik ini. Untuk deskripsi CloudWatch metrik yang tersedia untuk Amazon EMR, lihat metrik dan dimensi EMR Amazon di Panduan Pengguna Amazon. CloudWatch

Menggunakan AWS CLI untuk mengonfigurasi penskalaan otomatis

Anda dapat menggunakan perintah AWS CLI untuk Amazon EMR untuk mengonfigurasi penskalaan otomatis ketika Anda membuat sebuah klaster dan ketika Anda membuat grup instans. Anda dapat

menggunakan sintaks steno, menentukan konfigurasi JSON inline dalam perintah yang relevan, atau Anda dapat mereferensikan file yang berisi konfigurasi JSON. Anda juga dapat menerapkan kebijakan penskalaan otomatis ke grup instans yang ada dan menghapus kebijakan penskalaan otomatis yang sebelumnya diterapkan. Selain itu, Anda dapat mengambil detail konfigurasi kebijakan penskalaan dari klaster berjalan.

> **Important**
>
> Saat membuat klaster yang memiliki kebijakan penskalaan otomatis, Anda harus menggunakan `--auto-scaling-role` *`MyAutoScalingRole`* perintah tersebut untuk menentukan peran IAM untuk penskalaan otomatis. Peran default adalah *`EMR_AutoScaling_DefaultRole`* dan dapat dibuat dengan perintah `create-default-roles`. Peran hanya dapat ditambahkan ketika klaster dibuat, dan tidak dapat ditambahkan ke klaster yang ada.

Untuk penjelasan mendetail tentang parameter yang tersedia saat mengonfigurasi kebijakan penskalaan otomatis, lihat di Referensi PutAutoScalingPolicyAPI EMR Amazon.

Membuat sebuah klaster dengan kebijakan penskalaan otomatis diterapkan ke grup instans

Anda dapat menentukan konfigurasi penskalaan otomatis dalam opsi `--instance-groups` dari perintah `aws emr create-cluster`. Contoh berikut menggambarkan perintah create-cluster dimana kebijakan penskalaan otomatis untuk grup instans inti disediakan secara inline. Perintah membuat konfigurasi penskalaan yang setara dengan kebijakan penskalaan default yang muncul saat Anda membuat kebijakan penskalaan otomatis dengan EMR untuk AmazonAWS Management Console. Singkatnya, kebijakan penskalaan ke dalam tidak ditampilkan. Kami tidak menyarankan untuk membuat aturan penskalaan keluar tanpa aturan penskalaan ke dalam.

```
aws emr create-cluster --release-label emr-5.2.0 --service-role
 EMR_DefaultRole --ec2-attributes InstanceProfile=EMR_EC2_DefaultRole
 --auto-scaling-role EMR_AutoScaling_DefaultRole  --instance-groups
 Name=MyMasterIG,InstanceGroupType=MASTER,InstanceType=m5.xlarge,InstanceCount=1
 'Name=MyCoreIG,InstanceGroupType=CORE,InstanceType=m5.xlarge,InstanceCount=2,AutoScalingPolicy
scale-out,Description=Replicates the default scale-out rule in the
 console.,Action={SimpleScalingPolicyConfiguration={AdjustmentType=CHANGE_IN_CAPACITY,ScalingAd
ElasticMapReduce,Period=300,Statistic=AVERAGE,Threshold=15,Unit=PERCENT,Dimensions=[{Key=JobFlo
```

Perintah berikut mengilustrasikan cara menggunakan baris perintah untuk memberikan definisi kebijakan penskalaan otomatis sebagai bagian dari file konfigurasi grup instance bernama. *instancegroupconfig.json*

```
aws emr create-cluster --release-label emr-5.2.0 --service-role EMR_DefaultRole --ec2-attributes InstanceProfile=EMR_EC2_DefaultRole --instance-groups file://your/path/to/instancegroupconfig.json --auto-scaling-role EMR_AutoScaling_DefaultRole
```

Dengan isi file konfigurasi sebagai berikut:

```
[
{
  "InstanceCount": 1,
  "Name": "MyMasterIG",
  "InstanceGroupType": "MASTER",
  "InstanceType": "m5.xlarge"
},
{
  "InstanceCount": 2,
  "Name": "MyCoreIG",
  "InstanceGroupType": "CORE",
  "InstanceType": "m5.xlarge",
  "AutoScalingPolicy":
    {
     "Constraints":
      {
       "MinCapacity": 2,
       "MaxCapacity": 10
      },
     "Rules":
     [
      {
       "Name": "Default-scale-out",
       "Description": "Replicates the default scale-out rule in the console for YARN
 memory.",
       "Action":{
        "SimpleScalingPolicyConfiguration":{
          "AdjustmentType": "CHANGE_IN_CAPACITY",
          "ScalingAdjustment": 1,
          "CoolDown": 300
        }
       },
```

```
       "Trigger":{
        "CloudWatchAlarmDefinition":{
          "ComparisonOperator": "LESS_THAN",
          "EvaluationPeriods": 1,
          "MetricName": "YARNMemoryAvailablePercentage",
          "Namespace": "AWS/ElasticMapReduce",
          "Period": 300,
          "Threshold": 15,
          "Statistic": "AVERAGE",
          "Unit": "PERCENT",
          "Dimensions":[
             {
               "Key" : "JobFlowId",
               "Value" : "${emr.clusterId}"
             }
          ]
        }
       }
      }
     ]
   }
}
]
```

Menambahkan grup instans dengan kebijakan penskalaan otomatis ke klaster

Anda dapat menentukan konfigurasi kebijakan penskalaan dengan `--instance-groups` opsi dengan `add-instance-groups` perintah dengan cara yang sama seperti yang Anda bisa saat menggunakan`create-cluster`. Contoh berikut ini menggunakan referensi ke file JSON, *`instancegroupconfig.json`*, dengan konfigurasi grup instans.

```
aws emr add-instance-groups --cluster-id j-1EKZ3TYEVF1S2 --instance-groups file://your/
path/to/instancegroupconfig.json
```

Menerapkan kebijakan penskalaan otomatis ke grup instans yang ada atau memodifikasi suatu kebijakan yang diterapkan

Gunakan perintah `aws emr put-auto-scaling-policy` untuk menerapkan kebijakan penskalaan otomatis ke grup instans yang sudah ada. Grup instans harus menjadi bagian dari sebuah klaster yang menggunakan IAM role penskalaan otomatis. Contoh berikut ini menggunakan referensi ke file JSON, *`autoscaleconfig.json`*, yang menentukan konfigurasi kebijakan penskalaan otomatis.

```
aws emr put-auto-scaling-policy --cluster-id j-1EKZ3TYEVF1S2 --instance-group-
id ig-3PLUZBA6WLS07 --auto-scaling-policy file://your/path/to/autoscaleconfig.json
```

Isi dari file `autoscaleconfig.json`, yang mendefinisikan aturan penskalaan keluar yang sama seperti yang ditunjukkan pada contoh sebelumnya, ditunjukkan di bawah ini.

```
{
          "Constraints": {
                  "MaxCapacity": 10,
                  "MinCapacity": 2
          },
          "Rules": [{
                  "Action": {
                          "SimpleScalingPolicyConfiguration": {
                                  "AdjustmentType": "CHANGE_IN_CAPACITY",
                                  "CoolDown": 300,
                                  "ScalingAdjustment": 1
                          }
                  },
                  "Description": "Replicates the default scale-out rule in the console
 for YARN memory",
                  "Name": "Default-scale-out",
                  "Trigger": {
                          "CloudWatchAlarmDefinition": {
                                  "ComparisonOperator": "LESS_THAN",
                                  "Dimensions": [{
                                          "Key": "JobFlowID",
                                          "Value": "${emr.clusterID}"
                                  }],
                                  "EvaluationPeriods": 1,
                                  "MetricName": "YARNMemoryAvailablePercentage",
                                  "Namespace": "AWS/ElasticMapReduce",
                                  "Period": 300,
                                  "Statistic": "AVERAGE",
                                  "Threshold": 15,
                                  "Unit": "PERCENT"
                          }
                  }
          }]
 }
```

Menghapus kebijakan penskalaan otomatis dari grup instans

```
aws emr remove-auto-scaling-policy --cluster-id j-1EKZ3TYEVF1S2 --instance-group-
id ig-3PLUZBA6WLS07
```

Mengambil Konfigurasi Kebijakan Penskalaan Otomatis

`describe-cluster`Perintah mengambil konfigurasi kebijakan di InstanceGroup blok. Sebagai contoh, perintah berikut ini mengambil konfigurasi untuk klaster dengan ID klaster `j-1CWOHP4PI30VJ`.

```
aws emr describe-cluster --cluster-id j-1CWOHP4PI30VJ
```

Perintah tersebut menghasilkan output seperti berikut ini.

```
{
    "Cluster": {
        "Configurations": [],
        "Id": "j-1CWOHP4PI30VJ",
        "NormalizedInstanceHours": 48,
        "Name": "Auto Scaling Cluster",
        "ReleaseLabel": "emr-5.2.0",
        "ServiceRole": "EMR_DefaultRole",
        "AutoTerminate": false,
        "TerminationProtected": true,
        "MasterPublicDnsName": "ec2-54-167-31-38.compute-1.amazonaws.com",
        "LogUri": "s3n://aws-logs-232939870606-us-east-1/elasticmapreduce/",
        "Ec2InstanceAttributes": {
            "Ec2KeyName": "performance",
            "AdditionalMasterSecurityGroups": [],
            "AdditionalSlaveSecurityGroups": [],
            "EmrManagedSlaveSecurityGroup": "sg-09fc9362",
            "Ec2AvailabilityZone": "us-east-1d",
            "EmrManagedMasterSecurityGroup": "sg-0bfc9360",
            "IamInstanceProfile": "EMR_EC2_DefaultRole"
        },
        "Applications": [
            {
                "Name": "Hadoop",
```

```
                "Version": "2.7.3"
            }
        ],
        "InstanceGroups": [
            {
                "AutoScalingPolicy": {
                    "Status": {
                        "State": "ATTACHED",
                        "StateChangeReason": {
                            "Message": ""
                        }
                    },
                    "Constraints": {
                        "MaxCapacity": 10,
                        "MinCapacity": 2
                    },
                    "Rules": [
                        {
                            "Name": "Default-scale-out",
                            "Trigger": {
                                "CloudWatchAlarmDefinition": {
                                    "MetricName": "YARNMemoryAvailablePercentage",
                                    "Unit": "PERCENT",
                                    "Namespace": "AWS/ElasticMapReduce",
                                    "Threshold": 15,
                                    "Dimensions": [
                                        {
                                            "Key": "JobFlowId",
                                            "Value": "j-1CWOHP4PI30VJ"
                                        }
                                    ],
                                    "EvaluationPeriods": 1,
                                    "Period": 300,
                                    "ComparisonOperator": "LESS_THAN",
                                    "Statistic": "AVERAGE"
                                }
                            },
                            "Description": "",
                            "Action": {
                                "SimpleScalingPolicyConfiguration": {
                                    "CoolDown": 300,
                                    "AdjustmentType": "CHANGE_IN_CAPACITY",
                                    "ScalingAdjustment": 1
                                }
```

```
                        }
                    },
                    {
                        "Name": "Default-scale-in",
                        "Trigger": {
                            "CloudWatchAlarmDefinition": {
                                "MetricName": "YARNMemoryAvailablePercentage",
                                "Unit": "PERCENT",
                                "Namespace": "AWS/ElasticMapReduce",
                                "Threshold": 75,
                                "Dimensions": [
                                    {
                                        "Key": "JobFlowId",
                                        "Value": "j-1CWOHP4PI30VJ"
                                    }
                                ],
                                "EvaluationPeriods": 1,
                                "Period": 300,
                                "ComparisonOperator": "GREATER_THAN",
                                "Statistic": "AVERAGE"
                            }
                        },
                        "Description": "",
                        "Action": {
                            "SimpleScalingPolicyConfiguration": {
                                "CoolDown": 300,
                                "AdjustmentType": "CHANGE_IN_CAPACITY",
                                "ScalingAdjustment": -1
                            }
                        }
                    }
                ]
            },
            "Configurations": [],
            "InstanceType": "m5.xlarge",
            "Market": "ON_DEMAND",
            "Name": "Core - 2",
            "ShrinkPolicy": {},
            "Status": {
                "Timeline": {
                    "CreationDateTime": 1479413437.342,
                    "ReadyDateTime": 1479413864.615
                },
                "State": "RUNNING",
```

```
                "StateChangeReason": {
                    "Message": ""
                }
            },
            "RunningInstanceCount": 2,
            "Id": "ig-3M16XBE8C3PH1",
            "InstanceGroupType": "CORE",
            "RequestedInstanceCount": 2,
            "EbsBlockDevices": []
        },
        {
            "Configurations": [],
            "Id": "ig-OP62I28NSE8M",
            "InstanceGroupType": "MASTER",
            "InstanceType": "m5.xlarge",
            "Market": "ON_DEMAND",
            "Name": "Master - 1",
            "ShrinkPolicy": {},
            "EbsBlockDevices": [],
            "RequestedInstanceCount": 1,
            "Status": {
                "Timeline": {
                    "CreationDateTime": 1479413437.342,
                    "ReadyDateTime": 1479413752.088
                },
                "State": "RUNNING",
                "StateChangeReason": {
                    "Message": ""
                }
            },
            "RunningInstanceCount": 1
        }
    ],
    "AutoScalingRole": "EMR_AutoScaling_DefaultRole",
    "Tags": [],
    "BootstrapActions": [],
    "Status": {
        "Timeline": {
            "CreationDateTime": 1479413437.339,
            "ReadyDateTime": 1479413863.666
        },
        "State": "WAITING",
        "StateChangeReason": {
            "Message": "Cluster ready after last step completed."
```

```
            }
        }
    }
}
```

Secara manual mengubah ukuran klaster berjalan

Anda dapat menambahkan dan menghapus instance dari grup instance inti dan tugas serta armada instance di cluster yang sedang berjalan denganAWS Management Console,AWS CLI, atau Amazon EMR API. Jika klaster menggunakan grup instans, Anda secara eksplisit mengubah jumlah instans. Jika klaster Anda menggunakan armada instans, Anda dapat mengubah unit target untuk Instans Sesuai Permintaan dan Instans Spot. Armada instans lalu menambahkan dan menghapus instans untuk memenuhi target baru. Untuk informasi selengkapnya, lihat Opsi armada instans. Aplikasi dapat menggunakan instans Amazon EC2 yang baru disediakan untuk meng–host simpul segera setelah instans tersedia. Ketika instance dihapus, Amazon EMR menutup tugas dengan cara yang tidak mengganggu pekerjaan dan perlindungan terhadap kehilangan data. Untuk informasi selengkapnya, lihat Akhiri pada penyelesaian tugas.

Ubah ukuran cluster dengan konsol

Anda dapat menggunakan konsol Amazon EMR untuk mengubah ukuran klaster berjalan.

> **Note**
>
> Kami telah mendesain ulang konsol EMR Amazon agar lebih mudah digunakan. Lihat Apa yang baru dengan konsol? untuk mempelajari tentang perbedaan antara pengalaman konsol lama dan baru.

New console

Untuk mengubah jumlah instance untuk cluster yang ada dengan konsol baru

1. Masuk keAWS Management Console, dan buka konsol EMR Amazon di https://console.aws.amazon.com/emr.
2. Di bawah EMR pada EC2 di panel navigasi kiri, pilih Cluster, dan pilih cluster yang ingin Anda perbarui. Cluster harus berjalan; Anda tidak dapat mengubah ukuran klaster penyediaan atau terminasi.

3. Pada tab Instans pada halaman detail klaster, lihat panel grup Instans.
4. Untuk mengubah ukuran grup instans yang ada, pilih tombol radio di sebelah grup inti atau instance tugas yang ingin Anda ubah ukurannya, lalu pilih Ubah ukuran grup instans. Tentukan jumlah instance baru untuk grup instans, lalu pilih Ubah ukuran.

   > **Note**
   >
   > Jika Anda memilih untuk mengurangi ukuran grup instans yang sedang berjalan, Amazon EMR akan secara cerdas memilih instance yang akan dihapus dari grup untuk kehilangan data minimal. Untuk kontrol lebih terperinci dari tindakan mengubah ukuran Anda, Anda dapat memilih ID untuk grup instans, memilih instance yang ingin Anda hapus, dan kemudian menggunakan opsi Terminate. Untuk informasi lebih lanjut tentang perilaku penurunan skala cerdas, lihat. Menurunkan skala klaster

5. Jika Anda ingin membatalkan tindakan mengubah ukuran, Anda dapat memilih tombol radio untuk grup instans dengan status Mengubah ukuran dan kemudian memilih Berhenti mengubah ukuran dari tindakan daftar.
6. Untuk menambahkan satu atau beberapa grup instance tugas ke klaster Anda sebagai respons terhadap peningkatan beban kerja, pilih Tambahkan grup instans tugas dari tindakan daftar. Pilih jenis instans Amazon EC2, masukkan jumlah instance untuk grup tugas, lalu pilih Tambahkan grup instans tugas untuk kembali ke panel grup Instans untuk klaster Anda.

Old console

Untuk mengubah jumlah instance untuk cluster yang ada dengan konsol lama

1. Dari halaman Daftar Klaster, pilih sebuah klaster untuk diubah ukurannya.
2. Pada halaman Detail Klaster, pilih Perangkat Keras.
3. Jika klaster Anda menggunakan grup instans, pilih Ubah ukuran dalam kolom Jumlah instans kolom untuk grup instans yang ingin Anda ubah ukurannya, ketik jumlah instans baru, dan kemudian pilih tanda centang hijau.

   —ATAU—

   Jika klaster Anda menggunakan armada instans, pilih Ubah ukuran dalam kolom Kapasitas yang disediakan, ketik nilai baru untuk Unit Sesuai Permintaan dan Unit spot, lalu pilih Ubah ukuran.

Ketika Anda membuat perubahan pada jumlah simpul, Status grup instans akan diperbarui. Ketika perubahan yang Anda minta selesai, Status adalah berjalan.

Ubah ukuran cluster dengan AWS CLI

Anda dapat menggunakan AWS CLI untuk mengubah ukuran klaster berjalan. Anda dapat meningkatkan atau mengurangi jumlah simpul tugas, dan Anda dapat meningkatkan jumlah simpul inti dalam klaster berjalan. Dimungkinkan juga untuk mematikan instance di grup instance inti dengan AWS CLI atau API. Ini harus dilakukan dengan hati-hati. Mematikan instance di grup instance inti berisiko kehilangan data, dan instance tidak diganti secara otomatis.

Selain mengubah ukuran inti dan grup tugas, Anda juga dapat menambahkan satu atau beberapa grup instance tugas ke cluster yang sedang berjalan dengan. AWS CLI

Untuk mengubah ukuran cluster dengan mengubah jumlah instance dengan AWS CLI

Anda dapat menambahkan instance ke grup inti atau grup tugas, dan Anda dapat menghapus instance dari grup tugas dengan AWS CLI `modify-instance-groups` subperintah dengan parameter. `InstanceCount` Untuk menambahkan instans ke grup inti atau tugas, tingkatkan `InstanceCount`. Untuk mengurangi jumlah instans dalam grup tugas, kurangi nilai `InstanceCount`. Mengubah jumlah instans grup tugas ke 0 akan menghapus semua instans tetapi tidak menghapus grup instans tersebut.

- Untuk meningkatkan jumlah instans dalam grup instans tugas dari 3 ke 4, ketik perintah berikut ini dan ganti *ig-31JXXXXXXBT0* dengan ID grup instans.

  ```
  aws emr modify-instance-groups --instance-groups
   InstanceGroupId=ig-31JXXXXXXBT0,InstanceCount=4
  ```

  Untuk mengambil `InstanceGroupId`, gunakan subperintah `describe-cluster`. Outputnya adalah obyek JSON yang disebut `Cluster` yang berisi ID dari setiap grup instans. Untuk menggunakan perintah ini, Anda memerlukan ID cluster (yang dapat Anda ambil dengan `aws emr list-clusters` perintah atau konsol). Untuk mengambil ID grup instans, ketik perintah berikut ini dan ganti *j-2AXXXXXXGAPLF* dengan ID klaster.

  ```
  aws emr describe-cluster --cluster-id j-2AXXXXXXGAPLF
  ```

  DenganAWS CLI, Anda juga dapat menghentikan instance di grup instance inti dengan `--modify-instance-groups` subperintah.

> **Warning**
>
> Menentukan `EC2InstanceIdsToTerminate` harus dilakukan dengan hati-hati. Instans diakhiri segera, terlepas dari status aplikasi yang berjalan padanya, dan instans tidak secara otomatis diganti. Hal ini berlaku terlepas dari konfigurasi Perilaku menurunkan skala klaster tersebut. Mengakhiri sebuah instans dengan cara ini berisiko kehilangan data dan perilaku klaster tak terduga.

Untuk mengakhiri instans tertentu Anda memerlukan ID grup instans (dikembalikan oleh subperintah `aws emr describe-cluster --cluster-id`) dan ID instans (dikembalikan oleh subperintah `aws emr list-instances --cluster-id`), ketik perintah berikut, ganti *`ig-6RXXXXXX07SA`* dengan ID grup instans dan ganti *`i-f9XXXXf2`* dengan ID instans.

```
aws emr modify-instance-groups --instance-groups
 InstanceGroupId=ig-6RXXXXXX07SA,EC2InstanceIdsToTerminate=i-f9XXXXf2
```

Untuk informasi selengkapnya tentang menggunakan perintah Amazon EMR dalam AWS CLI, lihat https://docs.aws.amazon.com/cli/latest/reference/emr.

Untuk mengubah ukuran cluster dengan menambahkan grup instance tugas dengan AWS CLI

DenganAWS CLI, Anda dapat menambahkan dari 1-48 grup instance tugas ke cluster dengan subperintah. `--add-instance-groups` Grup instance tugas hanya dapat ditambahkan ke cluster yang berisi grup instance utama dan grup instance inti. Saat Anda menggunakanAWS CLI, Anda dapat menambahkan hingga lima grup instance tugas setiap kali Anda menggunakan `--add-instance-groups` subperintah.

1. Untuk menambahkan satu grup instans tugas ke klaster, ketik perintah berikut ini dan ganti *`j-JXBXXXXXX37R`* dengan ID klaster.

   ```
   aws emr add-instance-groups --cluster-id j-JXBXXXXXX37R --instance-groups
    InstanceCount=6,InstanceGroupType=task,InstanceType=m5.xlarge
   ```

2. Untuk menambahkan beberapa grup instans tugas ke klaster, ketik perintah berikut ini dan ganti *`j-JXBXXXXXX37R`* dengan ID klaster. Anda dapat menambahkan hingga lima grup instans tugas dalam satu perintah.

```
aws emr add-instance-groups --cluster-id j-JXBXXXXXX37R --instance-
groups InstanceCount=6,InstanceGroupType=task,InstanceType=m5.xlarge
 InstanceCount=10,InstanceGroupType=task,InstanceType=m5.xlarge
```

Untuk informasi selengkapnya tentang menggunakan perintah Amazon EMR dalam AWS CLI, lihat https://docs.aws.amazon.com/cli/latest/reference/emr.

Menginterupsi perubahan ukuran

Menggunakan Amazon EMR versi 4.1.0 atau yang lebih baru, Anda dapat mengeluarkan perubahan ukuran di tengah-tengah operasi perubahan ukuran yang sudah ada. Selain itu, Anda dapat menghentikan permintaan perubahan ukuran yang dikirimkan sebelumnya atau mengirimkan permintaan baru untuk menimpa permintaan sebelumnya tanpa menunggu hingga selesai. Anda juga dapat menghentikan pengubahan ukuran yang ada dari konsol atau dengan panggilan `ModifyInstanceGroups` API dengan jumlah saat ini sebagai jumlah target klaster.

Tangkapan layar berikut menunjukkan grup instans tugas yang diubah ukurannya tetapi dapat dihentikan dengan memilih Berhenti.

Untuk mengganggu pengubahan ukuran dengan AWS CLI

Anda dapat menggunakan AWS CLI untuk menghentikan pengubahan ukuran dengan `modify-instance-groups` subperintah. Asumsikan bahwa Anda memiliki enam instans dalam grup instans Anda dan Anda ingin meningkatkannya ke 10. Anda kemudian memutuskan bahwa Anda ingin membatalkan permintaan tersebut:

- Permintaan awal:

  ```
  aws emr modify-instance-groups --instance-groups
   InstanceGroupId=ig-myInstanceGroupId,InstanceCount=10
  ```

  Permintaan kedua untuk menghentikan permintaan pertama:

  ```
  aws emr modify-instance-groups --instance-groups
   InstanceGroupId=ig-myInstanceGroupId,InstanceCount=6
  ```

> **Note**
>
> Karena proses ini asinkron, Anda mungkin melihat jumlah instans berubah sehubungan dengan permintaan API sebelumnya sebelum permintaan berikutnya dituruti. Dalam kasus penyusutan, Anda mungkin mengalami ketika memiliki pekerjaan yang berjalan pada simpul, grup instans mungkin tidak menyusut sampai simpul telah menyelesaikan pekerjaan mereka.

Kondisi yang ditangguhkan

Grup instans memasuki status ditangguhkan jika menemui terlalu banyak kesalahan ketika mencoba untuk memulai simpul klaster baru. Sebagai contoh, jika simpul yang baru gagal saat melakukan tindakan bootstrap, grup instans masuk ke status DITANGGUHKAN, bukan terus-menerus menyediakan simpul baru. Setelah Anda mengatasi masalah yang mendasari, setel ulang jumlah simpul yang diinginkan pada grup instans di klaster, dan kemudian grup instans akan melanjutkan mengalokasikan simpul. Memodifikasi grup instans menginstruksikan Amazon EMR untuk mencoba menyediakan simpul kembali. Tidak ada simpul berjalan yang dimulai ulang atau dihentikan.

Di AWS CLI, subperintah `list-instances` mengembalikan semua instans dan statusnya seperti yang dilakukan subperintah `describe-cluster`. Jika Amazon EMR mendeteksi kesalahan dengan grup instans, itu akan mengubah status grup menjadi `SUSPENDED`.

Untuk mengatur ulang cluster dalam status SUSPEND dengan AWS CLI

Ketik subperintah `describe-cluster` dengan parameter `--cluster-id` untuk melihat status instans dalam klaster Anda.

- Untuk melihat informasi tentang semua instans dan grup instans dalam sebuah klaster, ketik perintah berikut dan ganti *`j-3KVXXXXXXY7UG`* dengan ID klaster.

  ```
  aws emr describe-cluster --cluster-id j-3KVXXXXXXY7UG
  ```

  Output menampilkan informasi tentang grup instans Anda dan status instans:

  ```
  {
      "Cluster": {
          "Status": {
              "Timeline": {
                  "ReadyDateTime": 1413187781.245,
  ```

```
                "CreationDateTime": 1413187405.356
            },
            "State": "WAITING",
            "StateChangeReason": {
                "Message": "Waiting after step completed"
            }
        },
        "Ec2InstanceAttributes": {
            "Ec2AvailabilityZone": "us-west-2b"
        },
        "Name": "Development Cluster",
        "Tags": [],
        "TerminationProtected": false,
        "RunningAmiVersion": "3.2.1",
        "NormalizedInstanceHours": 16,
        "InstanceGroups": [
            {
                "RequestedInstanceCount": 1,
                "Status": {
                    "Timeline": {
                        "ReadyDateTime": 1413187775.749,
                        "CreationDateTime": 1413187405.357
                    },
                    "State": "RUNNING",
                    "StateChangeReason": {
                        "Message": ""
                    }
                },
                "Name": "MASTER",
                "InstanceGroupType": "MASTER",
                "InstanceType": "m5.xlarge",
                "Id": "ig-3ETXXXXXXFYV8",
                "Market": "ON_DEMAND",
                "RunningInstanceCount": 1
            },
            {
                "RequestedInstanceCount": 1,
                "Status": {
                    "Timeline": {
                        "ReadyDateTime": 1413187781.301,
                        "CreationDateTime": 1413187405.357
                    },
                    "State": "RUNNING",
                    "StateChangeReason": {
```

```
                        "Message": ""
                    }
                },
                "Name": "CORE",
                "InstanceGroupType": "CORE",
                "InstanceType": "m5.xlarge",
                "Id": "ig-3SUXXXXXQ9ZM",
                "Market": "ON_DEMAND",
                "RunningInstanceCount": 1
            }
...
}
```

Untuk melihat informasi tentang grup instans tertentu, ketik subperintah `list-instances` dengan parameter `--cluster-id` dan `--instance-group-types`. Anda dapat melihat informasi untuk kelompok utama, inti, atau tugas.

```
aws emr list-instances --cluster-id j-3KVXXXXXXY7UG --instance-group-types "CORE"
```

Penggunaan subperintah `modify-instance-groups` dengan parameter `--instance-groups` untuk menyetel ulang klaster di status SUSPENDED. ID grup instans dikembalikan oleh subperintah `describe-cluster`.

```
aws emr modify-instance-groups --instance-groups
 InstanceGroupId=ig-3SUXXXXXQ9ZM,InstanceCount=3
```

Pertimbangan saat mengurangi ukuran cluster

Jika Anda memilih untuk mengurangi ukuran klaster yang sedang berjalan, pertimbangkan perilaku dan praktik terbaik EMR Amazon berikut:

- Untuk mengurangi dampak pada pekerjaan yang sedang berlangsung, Amazon EMR secara cerdas memilih instans yang akan dihapus. Untuk informasi selengkapnya tentang perilaku penurunan skala klaster, lihat Akhiri pada penyelesaian tugas di Panduan Manajemen EMR Amazon.
- Saat Anda menurunkan ukuran cluster, Amazon EMR menyalin data dari instance yang dihapus ke instance yang tersisa. Pastikan bahwa ada kapasitas penyimpanan yang cukup untuk data ini dalam kasus yang tetap dalam grup.

- Amazon EMR mencoba untuk menonaktifkan HDFS pada instance dalam grup. Sebelum Anda mengurangi ukuran cluster, kami sarankan Anda meminimalkan HDFS tulis I/O.
- Untuk kontrol yang paling terperinci ketika Anda mengurangi ukuran cluster, Anda dapat melihat cluster di konsol dan menavigasi ke tab Instances. Pilih ID untuk grup instance yang ingin Anda ubah ukurannya. Kemudian gunakan opsi Terminate untuk instance tertentu yang ingin Anda hapus.

Konfigurasikan batas waktu untuk kapasitas penyediaan

Saat menggunakan armada instance, Anda dapat mengonfigurasi batas waktu penyediaan. Batas waktu penyediaan menginstruksikan Amazon EMR untuk menghentikan penyediaan kapasitas instans jika klaster melebihi ambang waktu yang ditentukan selama peluncuran klaster atau operasi penskalaan klaster. Topik berikut mencakup cara mengonfigurasi batas waktu penyediaan untuk peluncuran klaster dan untuk operasi penskalaan klaster.

Topik

- Konfigurasikan batas waktu penyediaan untuk peluncuran klaster di Amazon EMR
- Kustomisasi periode batas waktu penyediaan untuk mengubah ukuran cluster di Amazon EMR

Konfigurasikan batas waktu penyediaan untuk peluncuran klaster di Amazon EMR

Anda dapat menentukan periode batas waktu untuk menyediakan Instans Spot untuk setiap armada di klaster Anda. Jika Amazon EMR tidak dapat menyediakan kapasitas Spot, Anda dapat memilih untuk menghentikan klaster atau menyediakan kapasitas Sesuai Permintaan. Jika periode batas waktu berakhir selama proses pengubahan ukuran klaster, Amazon EMR membatalkan permintaan Spot yang tidak tersedia. Instans Spot yang tidak disediakan tidak ditransfer ke kapasitas Sesuai Permintaan.

> **Note**
>
> Anda tidak dapat menyesuaikan periode batas waktu penyediaan di konsol lama. Lihat Apa yang baru dengan konsol? untuk mempelajari tentang perbedaan antara pengalaman konsol lama dan baru.

Lakukan langkah-langkah berikut untuk menyesuaikan periode batas waktu penyediaan untuk peluncuran klaster dengan konsol Amazon EMR.

New console

Untuk mengonfigurasi batas waktu penyediaan saat Anda membuat klaster dengan konsol baru

1. Masuk keAWS Management Console, dan buka konsol EMR Amazon di https://console.aws.amazon.com/emr.
2. Di bawah EMR pada EC2 di panel navigasi kiri, pilih Clusters, lalu pilih Create cluster.
3. Pada halaman Create Cluster, navigasikan ke konfigurasi Cluster dan pilih Instance Fleets.
4. Di bawah opsi penskalaan dan penyediaan klaster, tentukan ukuran Spot untuk inti dan armada tugas Anda.
5. Di bawah konfigurasi batas waktu Spot, pilih salah satu Kluster Terminate setelah batas waktu Spot atau Beralih ke Sesuai Permintaan setelah batas waktu Spot. Kemudian, tentukan periode batas waktu untuk penyediaan Instans Spot. Nilai default adalah 1 jam.
6. Pilih opsi lain yang berlaku untuk klaster Anda.
7. Untuk meluncurkan klaster Anda dengan batas waktu yang dikonfigurasi, pilih Buat klaster.

AWS CLI

Untuk menentukan batas waktu penyediaan dengan perintah **`create-cluster`**

```
aws emr create-cluster \
--release-label emr-5.35.0 \
--service-role EMR_DefaultRole \
--ec2-attributes '{"InstanceProfile":"EMR_EC2_DefaultRole","SubnetIds":["subnet-
XXXXX"]}' \
--instance-fleets
 '[{"InstanceFleetType":"MASTER","TargetOnDemandCapacity":1,"TargetSpotCapacity":0,"LaunchSp
{"OnDemandSpecification":{"AllocationStrategy":"lowest-
price"}},"InstanceTypeConfigs":[{"WeightedCapacity":1,"EbsConfiguration":
{"EbsBlockDeviceConfigs":[{"VolumeSpecification":
{"SizeInGB":32,"VolumeType":"gp2"},"VolumesPerInstance":2}]},"BidPriceAsPercentageOfOnDemand
 - 1"},
{"InstanceFleetType":"CORE","TargetOnDemandCapacity":1,"TargetSpotCapacity":1,"LaunchSpecifi
{"SpotSpecification":
{"TimeoutDurationMinutes":120,"TimeoutAction":"SWITCH_TO_ON_DEMAND"},"OnDemandSpecification"
{"AllocationStrategy":"lowest-price"}},"InstanceTypeConfigs":
[{"WeightedCapacity":1,"EbsConfiguration":{"EbsBlockDeviceConfigs":
[{"VolumeSpecification":
```

```
{"SizeInGB":32,"VolumeType":"gp2"},"VolumesPerInstance":2}]},"BidPriceAsPercentageOfOnDemand
 - 2"}]'
```

Kustomisasi periode batas waktu penyediaan untuk mengubah ukuran cluster di Amazon EMR

Anda dapat menentukan periode batas waktu untuk menyediakan Instans Spot untuk setiap armada di klaster Anda. Jika Amazon EMR tidak dapat menyediakan kapasitas Spot, Amazon akan membatalkan permintaan pengubahan ukuran dan menghentikan upayanya untuk menyediakan kapasitas Spot tambahan. Saat Anda membuat cluster, Anda dapat mengonfigurasi batas waktu. Untuk klaster yang sedang berjalan, Anda dapat menambahkan atau memperbarui batas waktu.

Ketika periode batas waktu berakhir, Amazon EMR secara otomatis mengirimkan acara ke aliran Acara Amazon. CloudWatch Dengan CloudWatch, Anda dapat membuat aturan yang cocok dengan peristiwa sesuai dengan pola yang ditentukan, dan kemudian merutekan peristiwa ke target untuk mengambil tindakan. Misalnya, Anda dapat mengonfigurasi aturan untuk mengirim pemberitahuan email. Untuk informasi selengkapnya tentang cara membuat aturan, lihatMembuat aturan untuk acara EMR Amazon dengan CloudWatch. Untuk informasi selengkapnya tentang detail acara yang berbeda, lihatInstance peristiwa perubahan negara armada.

Contoh batas waktu penyediaan untuk pengubahan ukuran klaster

Tentukan batas waktu penyediaan untuk mengubah ukuran dengan AWS CLI

Contoh berikut menggunakan `create-cluster` perintah untuk menambahkan batas waktu penyediaan untuk mengubah ukuran.

```
aws emr create-cluster \
--release-label emr-5.35.0 \
--service-role EMR_DefaultRole \
--ec2-attributes '{"InstanceProfile":"EMR_EC2_DefaultRole","SubnetIds":["subnet-
XXXXX"]}' \
--instance-fleets
 '[{"InstanceFleetType":"MASTER","TargetOnDemandCapacity":1,"TargetSpotCapacity":0,"InstanceTyp
[{"WeightedCapacity":1,"EbsConfiguration":{"EbsBlockDeviceConfigs":
[{"VolumeSpecification":
{"SizeInGB":32,"VolumeType":"gp2"},"VolumesPerInstance":2}]},"BidPriceAsPercentageOfOnDemandPri
 - 1"},
{"InstanceFleetType":"CORE","TargetOnDemandCapacity":1,"TargetSpotCapacity":1,"LaunchSpecificat
{"SpotSpecification":
```

```
{"TimeoutDurationMinutes":120,"TimeoutAction":"SWITCH_TO_ON_DEMAND"},"OnDemandSpecification":
{"AllocationStrategy":"lowest-price"}},"ResizeSpecifications":
{"SpotResizeSpecification":{"TimeoutDurationMinutes":20},"OnDemandResizeSpecification":
{"TimeoutDurationMinutes":25}},"InstanceTypeConfigs":
[{"WeightedCapacity":1,"EbsConfiguration":{"EbsBlockDeviceConfigs":
[{"VolumeSpecification":
{"SizeInGB":32,"VolumeType":"gp2"},"VolumesPerInstance":2}]},"BidPriceAsPercentageOfOnDemandPri
 - 2"}]'
```

Contoh berikut menggunakan `modify-instance-fleet` perintah untuk menambahkan batas waktu penyediaan untuk mengubah ukuran.

```
aws emr modify-instance-fleet \
--cluster-id j-XXXXXXXXXXXX \
--instance-fleet '{"InstanceFleetId":"if-XXXXXXXXXXX","ResizeSpecifications":
{"SpotResizeSpecification":{"TimeoutDurationMinutes":30},"OnDemandResizeSpecification":
{"TimeoutDurationMinutes":60}}}' \
--region us-east-1
```

Contoh berikut menggunakan `add-instance-fleet-command` untuk menambahkan batas waktu penyediaan untuk mengubah ukuran.

```
aws emr add-instance-fleet \
--cluster-id j-XXXXXXXXXXXX \
--instance-fleet
 '{"InstanceFleetType":"TASK","TargetOnDemandCapacity":1,"TargetSpotCapacity":0,"InstanceTypeCo
[{"WeightedCapacity":1,"EbsConfiguration":{"EbsBlockDeviceConfigs":
[{"VolumeSpecification":
{"SizeInGB":32,"VolumeType":"gp2"},"VolumesPerInstance":2}]},"BidPriceAsPercentageOfOnDemandPri
{"SpotResizeSpecification":{"TimeoutDurationMinutes":30},"OnDemandResizeSpecification":
{"TimeoutDurationMinutes":35}}}' \
--region us-east-1
```

Tentukan batas waktu penyediaan untuk mengubah ukuran dan peluncuran dengan AWS CLI

Contoh berikut menggunakan `create-cluster` perintah untuk menambahkan batas waktu penyediaan untuk mengubah ukuran dan peluncuran.

```
aws emr create-cluster \
--release-label emr-5.35.0 \
--service-role EMR_DefaultRole \
```

```
--ec2-attributes '{"InstanceProfile":"EMR_EC2_DefaultRole","SubnetIds":["subnet-
XXXXX"]}' \
--instance-fleets
 '[{"InstanceFleetType":"MASTER","TargetOnDemandCapacity":1,"TargetSpotCapacity":0,"LaunchSpeci
{"OnDemandSpecification":{"AllocationStrategy":"lowest-price"}},"InstanceTypeConfigs":
[{"WeightedCapacity":1,"EbsConfiguration":{"EbsBlockDeviceConfigs":
[{"VolumeSpecification":
{"SizeInGB":32,"VolumeType":"gp2"},"VolumesPerInstance":2}]},"BidPriceAsPercentageOfOnDemandPri
 - 1"},
{"InstanceFleetType":"CORE","TargetOnDemandCapacity":1,"TargetSpotCapacity":1,"LaunchSpecificat
{"SpotSpecification":
{"TimeoutDurationMinutes":120,"TimeoutAction":"SWITCH_TO_ON_DEMAND"},"OnDemandSpecification":
{"AllocationStrategy":"lowest-price"}},"ResizeSpecifications":
{"SpotResizeSpecification":{"TimeoutDurationMinutes":20},"OnDemandResizeSpecification":
{"TimeoutDurationMinutes":25}},"InstanceTypeConfigs":
[{"WeightedCapacity":1,"EbsConfiguration":{"EbsBlockDeviceConfigs":
[{"VolumeSpecification":
{"SizeInGB":32,"VolumeType":"gp2"},"VolumesPerInstance":2}]},"BidPriceAsPercentageOfOnDemandPri
 - 2"}]'
```

Pertimbangan untuk mengubah ukuran batas waktu penyediaan

Saat mengonfigurasi batas waktu penyediaan klaster untuk armada instans, pertimbangkan perilaku berikut.

- Anda dapat mengonfigurasi batas waktu penyediaan untuk Instans Spot dan Sesuai Permintaan. Batas waktu penyediaan minimum adalah 5 menit. Batas waktu penyediaan maksimum adalah 7 hari.
- Anda hanya dapat mengonfigurasi batas waktu penyediaan untuk klaster EMR yang menggunakan armada instance. Anda harus mengkonfigurasi setiap inti dan armada tugas secara terpisah.
- Saat membuat klaster, Anda dapat mengonfigurasi batas waktu penyediaan. Anda dapat menambahkan batas waktu atau memperbarui batas waktu yang ada untuk klaster yang sedang berjalan.
- Jika Anda mengirimkan beberapa operasi pengubahan ukuran, Amazon EMR melacak batas waktu penyediaan untuk setiap operasi pengubahan ukuran. *Misalnya, atur batas waktu penyediaan pada klaster menjadi 60 menit. Kemudian, kirimkan operasi pengubahan ukuran R1 pada waktu T1. Kirim operasi pengubahan ukuran kedua R2 pada waktu T2. Batas waktu penyediaan untuk R1 berakhir pada T1 + 60 menit. Batas waktu penyediaan untuk R2 berakhir pada T2 + 60 menit.*

- Jika Anda mengirimkan operasi pengubahan ukuran skala baru sebelum batas waktu berakhir, Amazon EMR melanjutkan upayanya untuk menyediakan kapasitas untuk klaster EMR Anda.

Menurunkan skala klaster

> **Note**
>
> Opsi perilaku penskalaan tidak lagi didukung sejak Amazon EMR merilis 5.10.0. Karena pengenalan penagihan per detik di Amazon EC2, perilaku menurunkan skala default untuk klaster Amazon EMR sekarang berakhir pada penyelesaian tugas.

Dengan Amazon EMR merilis 5.1.0 hingga 5.9.1, ada dua opsi untuk perilaku penurunan skala: berhenti pada batas jam instans untuk penagihan Amazon EC2, atau dihentikan saat penyelesaian tugas. Dimulai dengan rilis Amazon EMR 5.10.0, pengaturan untuk penghentian pada batas jam instans tidak digunakan lagi karena pengenalan penagihan per detik di Amazon EC2. Kami tidak merekomendasikan menentukan pengakhiran batas jam instans dalam versi yang memiliki opsi tersebut.

> **Warning**
>
> Jika Anda menggunakan AWS CLI untuk mengeluarkan `modify-instance-groups` dengan `EC2InstanceIdsToTerminate`, instans ini diakhiri segera, tanpa pertimbangan untuk pengaturan ini, dan mengesampingkan status aplikasi yang berjalan pada mereka. Mengakhiri sebuah instans dengan cara ini berisiko kehilangan data dan perilaku klaster tak terduga.

Saat penghentian saat penyelesaian tugas ditentukan, Amazon EMR menolak daftar dan menguras tugas dari node sebelum menghentikan instans Amazon EC2. Dengan ditentukannya salah satu perilaku tersebut, Amazon EMR tidak mengakhiri instans Amazon EC2 dalam grup instans inti jika dapat menyebabkan kerusakan HDFS.

Akhiri pada penyelesaian tugas

Amazon EMR mengizinkan Anda untuk menurunkan skala klaster Anda tanpa mempengaruhi beban kerja Anda. Amazon EMR menonaktifkan daemon YARN, HDFS, dan daemon lainnya pada simpul inti dan tugas selama menurunkan ukuran operasi tanpa kehilangan data atau mengganggu

pekerjaan. Amazon EMR hanya mengurangi ukuran grup instans jika pekerjaan yang ditetapkan ke grup telah selesai dan tidak digunakan. Untuk NodeManager YARN Graceful Decommission, Anda dapat secara manual menyesuaikan waktu node menunggu untuk dinonaktifkan.

Kali ini diatur menggunakan properti di klasifikasi konfigurasi `YARN-site`. Menggunakan Amazon EMR rilis 5.12.0 dan yang lebih tinggi, tentukan properti. `YARN.resourcemanager.nodemanager-graceful-decommission-timeout-secs` Menggunakan rilis Amazon EMR sebelumnya, tentukan properti. `YARN.resourcemanager.decommissioning.timeout`

Jika masih ada kontainer atau aplikasi YARN yang berjalan saat waktu penonaktifan habis, simpul dipaksa untuk dinonaktifkan dan YARN menjadwalkan ulang kontainer yang terpengaruh pada simpul lainnya. Nilai default adalah 3600 detik (satu jam). Anda dapat mengatur batas waktu ini menjadi nilai tinggi yang sewenang-wenang untuk memaksa pengurangan anggun menunggu lebih lama. Untuk informasi lebih lanjut, lihat Graceful Decommission of YARN nodes dalam dokumentasi Apache Hadoop.

Grup simpul tugas

Amazon EMR secara cerdas memilih instance yang tidak memiliki tugas yang berjalan terhadap langkah atau aplikasi apa pun, dan menghapus instance tersebut dari cluster terlebih dahulu. Jika semua instance di cluster sedang digunakan, Amazon EMR menunggu tugas diselesaikan pada instance sebelum menghapusnya dari cluster. Waktu tunggu default adalah 1 jam. Nilai ini dapat diubah dengan `YARN.resourcemanager.decommissioning.timeout` pengaturan. Amazon EMR secara dinamis menggunakan pengaturan baru. Anda dapat menyetel ini ke jumlah besar yang sewenang-wenang untuk memastikan bahwa Amazon EMR tidak menghentikan tugas apa pun sekaligus mengurangi ukuran cluster.

Grup simpul inti

Pada node inti, DataNode daemon YARN NodeManager dan HDFS harus dinonaktifkan agar grup instance dapat dikurangi. Untuk YARN, pengurangan anggun memastikan bahwa node yang ditandai untuk penonaktifan hanya dialihkan ke `DECOMMISSIONED` status jika tidak ada wadah atau aplikasi yang tertunda atau tidak lengkap. Penonaktifan segera selesai jika tidak ada kontainer yang berjalan pada simpul di awal penonaktifan.

Untuk HDFS, pengurangan yang anggun memastikan bahwa kapasitas target HDFS cukup besar untuk memenuhi semua blok yang ada. Jika kapasitas target tidak cukup besar, hanya sebagian jumlah instans inti yang dinonaktifkan sehingga simpul yang tersisa dapat menangani data yang ada di HDFS. Anda harus memastikan kapasitas HDFS tambahan untuk memungkinkan penonaktifan

lebih lanjut. Anda juga harus mencoba meminimalkan penulisan I/O sebelum mencoba mengurangi grup instance. I/O tulis yang berlebihan mungkin menunda penyelesaian operasi pengubahan ukuran.

Batas lain adalah faktor replikasi default, `dfs.replication` di dalam `/etc/hadoop/conf/hdfs-site`. Saat membuat cluster, Amazon EMR mengonfigurasi nilai berdasarkan jumlah instance di cluster: 1 dengan 1-3 instance, untuk cluster dengan 4-9 instance, dan 2 untuk cluster dengan 10+ instance. 3

Warning

1. Pengaturan `dfs.replication` ke 1 pada cluster dengan kurang dari empat node dapat menyebabkan hilangnya data HDFS jika satu node turun. Kami menyarankan Anda menggunakan cluster dengan setidaknya empat node inti untuk beban kerja produksi.
2. Amazon EMR tidak akan mengizinkan cluster untuk menskalakan node inti di bawah ini. `dfs.replication` Misalnya, jika`dfs.replication = 2`, jumlah minimum node inti adalah 2.
3. Saat Anda menggunakan Penskalaan Terkelola, Penskalaan Otomatis, atau memilih untuk mengubah ukuran klaster secara manual, sebaiknya atur `dfs.replication` ke 2 atau lebih tinggi.

Pengurangan yang anggun tidak memungkinkan Anda mengurangi node inti di bawah faktor replikasi HDFS. Ini untuk memungkinkan HDFS menutup file karena replika tidak mencukupi. Untuk menghindari batas ini, turunkan faktor replikasi dan restart daemon. NameNode

Mengonfigurasi perilaku menurunkan skala Amazon EMR

Note

Opsi perilaku scale-down terminate at instance hour tidak lagi didukung untuk rilis Amazon EMR 5.10.0 dan yang lebih tinggi. Opsi perilaku penurunan skala berikut hanya muncul di konsol EMR Amazon untuk rilis 5.1.0 hingga 5.9.1.

Anda dapat menggunakan AWS Management Console, AWS CLI, atau API Amazon EMR untuk mengonfigurasi perilaku menurunkan skala saat Anda membuat sebuah klaster.

> **Note**
>
> Kami telah mendesain ulang konsol EMR Amazon agar lebih mudah digunakan. Lihat Apa yang baru dengan konsol? untuk mempelajari tentang perbedaan antara pengalaman konsol lama dan baru.

New console

Untuk mengonfigurasi perilaku penurunan skala dengan konsol baru

1. Masuk keAWS Management Console, dan buka konsol EMR Amazon di https://console.aws.amazon.com/emr.
2. Di bawah EMR pada EC2 di panel navigasi kiri, pilih Clusters, lalu pilih Create cluster.
3. Di bagian opsi penskalaan dan penyediaan klaster, temukan Pengakhiran klaster dan pilih untuk menghentikan klaster Anda secara manual atau meminta Amazon EMR menghentikan klaster Anda setelah jumlah waktu idle yang ditentukan. Secara opsional, aktifkan perlindungan penghentian terhadap bug atau kesalahan.
4. Pilih opsi lain yang berlaku untuk cluster Anda.
5. Untuk meluncurkan klaster Anda, pilih Buat klaster.

Old console

Untuk mengonfigurasi perilaku penskalaan dengan konsol lama

1. Buka konsol Amazon EMR. di https://console.aws.amazon.com/elasticmapreduce.
2. Pilih Buat klaster. Buka Opsi lanjutan dan pilih pengaturan konfigurasi Anda di Langkah 1: Perangkat Lunak dan Langkah dan Langkah 2: Perangkat Keras.
3. Pada Langkah 3: Pengaturan Cluster Umum, pilih perilaku penskalaan yang Anda inginkan. Selesaikan konfigurasi yang tersisa dan buat cluster Anda.

AWS CLI

Untuk mengonfigurasi perilaku penskalaan dengan AWS CLI

- Gunakan `--scale-down-behavior` opsi untuk menentukan salah satu `TERMINATE_AT_INSTANCE_HOUR` atau`TERMINATE_AT_TASK_COMPLETION`.

Mengakhiri suatu klaster

Bagian ini menjelaskan metode untuk mengakhiri klaster. Untuk informasi tentang mengaktifkan proteksi pengakhiran dan mengakhiri klaster secara otomatis, lihat Pengakhiran kontrol klaster. Anda dapat mengakhiri klaster dalam status `STARTING`, `RUNNING`, atau `WAITING`. Klaster dalam status `WAITING` harus diakhiri atau ia akan berjalan selamanya, sehingga menimbulkan biaya ke akun Anda. Anda dapat mengakhiri sebuah klaster yang gagal untuk meninggalkan status `STARTING` atau tidak dapat menyelesaikan suatu langkah.

Jika Anda ingin mengakhiri klaster yang memiliki perlindungan terminasi yang disetel di atasnya, Anda harus menonaktifkan perlindungan terminasi sebelum Anda dapat mengakhiri klaster. Klaster dapat diakhiri dengan menggunakan konsol tersebut, AWS CLI, atau secara terprogram menggunakan `TerminateJobFlows` API.

Bergantung pada konfigurasi cluster, dibutuhkan waktu 5 hingga 20 menit bagi cluster untuk sepenuhnya menghentikan dan melepaskan sumber daya yang dialokasikan, seperti instans EC2.

> **Note**
>
> Anda tidak dapat me-restart klaster yang diakhiri, tetapi Anda dapat mengkloning klaster yang diakhiri untuk menggunakan kembali konfigurasinya untuk klaster baru. Untuk informasi selengkapnya, lihat Meng-klon klaster menggunakan konsol.

> **Important**
>
> Amazon EMR menggunakan peran layanan EMR Amazon dan `AWSServiceRoleForEMRCleanup` peran untuk membersihkan sumber daya klaster di akun yang tidak lagi Anda gunakan, seperti instans Amazon EC2. Anda harus menyertakan tindakan agar kebijakan peran menghapus atau menghentikan sumber daya. Jika tidak, Amazon EMR tidak dapat melakukan tindakan pembersihan ini, dan Anda mungkin dikenakan biaya untuk sumber daya yang tidak digunakan yang tetap ada di klaster.

Mengakhiri cluster dengan konsol

Anda dapat mengakhiri satu atau lebih klaster menggunakan konsol Amazon EMR. Langkah-langkah untuk mengakhiri sebuah klaster di konsol bervariasi tergantung pada apakah proteksi pengakhiran

aktif atau tidak. Untuk mengakhiri klaster yang diproteksi, Anda harus terlebih dahulu menonaktifkan proteksi pengakhiran.

New console

Untuk mengakhiri cluster dengan konsol baru

1. Masuk keAWS Management Console, dan buka konsol EMR Amazon di https://console.aws.amazon.com/emr.
2. Pilih Cluster, lalu pilih cluster yang ingin Anda akhiri.
3. Di bawah menu tarik-turun Tindakan, pilih Terminate cluster untuk membuka prompt Kluster Terminate.
4. Pada prompt, pilih Hentikan. Tergantung pada konfigurasi cluster, penghentian mungkin memakan waktu 5 hingga 10 menit. Untuk informasi selengkapnya tentang cara menggunakan kluster EMR Amazon, lihat. Mengakhiri suatu klaster

Old console

Untuk mengakhiri cluster dengan perlindungan terminasi mati dengan konsol lama

1. Arahkan ke konsol EMR Amazon baru dan pilih Beralih ke konsol lama dari navigasi samping. Untuk informasi selengkapnya tentang apa yang diharapkan saat beralih ke konsol lama, lihat Menggunakan konsol lama.
2. Pilih klaster untuk diakhiri. Anda dapat memilih beberapa klaster dan mengakhirinya pada waktu yang sama.
3. Pilih Akhiri.
4. Saat diminta, pilih Akhiri.

   Amazon EMR mengakhiri instans pada klaster dan berhenti menyimpan data log.

Untuk mengakhiri cluster dengan perlindungan terminasi aktif dengan konsol lama

1. Arahkan ke konsol EMR Amazon baru dan pilih Beralih ke konsol lama dari navigasi samping. Untuk informasi selengkapnya tentang apa yang diharapkan saat beralih ke konsol lama, lihat Menggunakan konsol lama.
2. Pada halaman Daftar Klaster, pilih klaster untuk diakhiri. Anda dapat memilih beberapa klaster dan mengakhirinya pada waktu yang sama.

3. Pilih Akhiri.
4. Saat diminta, pilih Ubah untuk menonaktifkan proteksi pengakhiran. Jika Anda memilih beberapa klaster, pilih Matikan semua untuk menonaktifkan proteksi pengakhiran untuk semua klaster sekaligus.
5. Pada dialog Akhiri klaster, untuk Proteksi Pengakhiran, pilih Mati dan kemudian klik tanda centang untuk mengkonfirmasi.
6. Klik Akhiri.

   Amazon EMR mengakhiri instans pada klaster dan berhenti menyimpan data log.

Mengakhiri cluster dengan AWS CLI

Untuk mengakhiri klaster yang tidak terlindungi menggunakan AWS CLI

Untuk mengakhiri klaster yang tidak terlindungi menggunakan AWS CLI, gunakan subperintah `terminate-clusters` dengan parameter --cluster-ids.

- Ketik perintah berikut untuk mengakhiri satu klaster dan mengganti *j-3KVXXXXXXX7UG* dengan ID klaster Anda.

  ```
  aws emr terminate-clusters --cluster-ids j-3KVXXXXXXX7UG
  ```

  Untuk mengakhiri beberapa klaster, ketik perintah berikut dan ganti *j-3KVXXXXXXX7UG* dan *j-WJ2XXXXXX8EU* dengan ID klaster Anda.

  ```
  aws emr terminate-clusters --cluster-ids j-3KVXXXXXXX7UG j-WJ2XXXXXX8EU
  ```

  Untuk informasi selengkapnya tentang menggunakan perintah Amazon EMR dalam AWS CLI, lihat https://docs.aws.amazon.com/cli/latest/reference/emr.

Untuk mengakhiri klaster yang dilindungi menggunakan AWS CLI

Untuk mengakhiri klaster yang dilindungi menggunakan AWS CLI, pertama-tama nonaktifkan proteksi pengakhiran menggunakan subperintah `modify-cluster-attributes` dengan parameter `--no-termination-protected`. Kemudian gunakan subperintah `terminate-clusters` dengan parameter `--cluster-ids` untuk mengakhirinya.

1. Ketik perintah berikut untuk menonaktifkan proteksi pengakhiran dan ganti *j-3KVTXXXXXX7UG* dengan ID klaster Anda.

   ```
   aws emr modify-cluster-attributes --cluster-id j-3KVTXXXXXX7UG --no-termination-protected
   ```

2. Untuk mengakhiri klaster, ketik perintah berikut dan ganti *j-3KVXXXXXXX7UG* dengan ID klaster Anda.

   ```
   aws emr terminate-clusters --cluster-ids j-3KVXXXXXXX7UG
   ```

   Untuk mengakhiri beberapa klaster, ketik perintah berikut dan ganti *j-3KVXXXXXXX7UG* dan *j-WJ2XXXXXX8EU* dengan ID klaster Anda.

   ```
   aws emr terminate-clusters --cluster-ids j-3KVXXXXXXX7UG j-WJ2XXXXXX8EU
   ```

   Untuk informasi selengkapnya tentang menggunakan perintah Amazon EMR dalam AWS CLI, lihat https://docs.aws.amazon.com/cli/latest/reference/emr.

Mengakhiri cluster dengan API

Operasi `TerminateJobFlows` menghentikan pengolahan langkah, mengunggah setiap data log dari Amazon EC2 ke Amazon S3 (jika dikonfigurasi), dan mengakhiri klaster Hadoop. Sebuah klaster juga berakhir secara otomatis jika Anda mengatur `KeepJobAliveWhenNoSteps` ke `False` dalam permintaan `RunJobFlows`.

Anda dapat menggunakan tindakan ini untuk mengakhiri satu klaster atau daftar klaster dengan ID klaster mereka.

Untuk informasi selengkapnya tentang parameter input yang unik`TerminateJobFlows`, lihat TerminateJobFlows. Untuk informasi selengkapnya tentang parameter generik dalam permintaan, lihat Parameter permintaan umum.

Meng-klon klaster menggunakan konsol

Anda dapat menggunakan konsol Amazon EMR untuk meng-klon sebuah klaster, yang membuat salinan konfigurasi klaster asli untuk digunakan sebagai dasar untuk klaster baru.

> **Note**
>
> Kami telah mendesain ulang konsol EMR Amazon agar lebih mudah digunakan. Anda dapat mengkloning cluster yang menggunakan penskalaan otomatis di konsol baru, tetapi Anda hanya dapat membuat klaster baru jika Anda ingin menskalakannya secara manual atau menggunakan penskalaan terkelola. Lihat Apa yang baru dengan konsol? untuk mempelajari lebih lanjut tentang perbedaan antara pengalaman konsol lama dan baru.

New console

Untuk mengkloning cluster dengan konsol baru

1. Masuk keAWS Management Console, dan buka konsol EMR Amazon di https://console.aws.amazon.com/emr.
2. Di bawah EMR pada EC2 di panel navigasi kiri, pilih Clusters.
3. Untuk mengkloning cluster dari daftar cluster
   a. Gunakan opsi pencarian dan filter untuk menemukan cluster yang ingin Anda kloning dalam tampilan daftar.
   b. Pilih kotak centang di sebelah kiri baris untuk cluster yang ingin Anda kloning.
   c. Opsi Clone sekarang akan tersedia di bagian atas tampilan daftar. Pilih Clone untuk memulai proses kloning. Jika klaster memiliki langkah-langkah yang dikonfigurasi, pilih Sertakan langkah dan Lanjutkan jika Anda ingin mengkloning langkah-langkah bersama dengan konfigurasi cluster lainnya.
   d. Tinjau pengaturan untuk cluster baru yang telah disalin dari cluster kloning. Sesuaikan pengaturan jika diperlukan. Bila Anda puas dengan konfigurasi cluster baru, pilih Create cluster untuk meluncurkan cluster baru.
4. Untuk mengkloning cluster dari halaman detail cluster
   a. Untuk menavigasi ke halaman detail klaster yang ingin Anda kloning, pilih ID Cluster-nya dari tampilan daftar cluster.
   b. Di bagian atas halaman detail cluster, pilih Clone cluster dari menu Actions untuk memulai proses kloning. Jika klaster memiliki langkah-langkah yang dikonfigurasi, pilih Sertakan langkah dan Lanjutkan jika Anda ingin mengkloning langkah-langkah bersama dengan konfigurasi cluster lainnya.

c. Tinjau pengaturan untuk cluster baru yang telah disalin dari cluster kloning. Sesuaikan pengaturan jika diperlukan. Bila Anda puas dengan konfigurasi cluster baru, pilih Create cluster untuk meluncurkan cluster baru.

Old console

Untuk mengkloning cluster menggunakan konsol lama

1. Arahkan ke konsol EMR Amazon baru dan pilih Beralih ke konsol lama dari navigasi samping. Untuk informasi selengkapnya tentang apa yang diharapkan saat beralih ke konsol lama, lihat Menggunakan konsol lama.
2. Pilih Buat klaster.
3. Dari halaman Daftar Klaster, klik sebuah klaster untuk di-klon.
4. Di bagian atas halaman Detail Klaster, klik Klon.

   Dalam kotak dialog, pilih Ya untuk memasukkan langkah-langkah dari klaster asli ke dalam klaster hasil klon. Pilih Tidak untuk meng-klon konfigurasi klaster asli tanpa termasuk salah satu langkah.

   > **Note**
   >
   > Untuk klaster yang dibuat menggunakan AMI 3.1.1 dan yang lebih baru (Hadoop 2.x) atau AMI 2.4.8 dan yang lebih baru (Hadoop 1.x), jika Anda meng-klon klaster dan menyertakan langkah-langkah, semua langkah-langkah sistem (seperti mengkonfigurasi Hive) di-klon bersama dengan langkah-langkah yang dikirim pengguna, hingga 1.000 total. Langkah lama yang tidak lagi muncul di riwayat langkah konsol tidak dapat di-klon. Untuk AMI sebelumnya, hanya 256 langkah yang dapat di-klo (termasuk langkah-langkah sistem). Untuk informasi selengkapnya, lihat Kirim pekerjaan ke sebuah klaster.
5. Halaman Buat Klaster muncul dengan salinan konfigurasi klaster asli. Tinjau konfigurasi, buat perubahan yang diperlukan, dan kemudian klik Buat Klaster.

Mengotomatisasi klaster berulang dengan AWS Data Pipeline

AWS Data Pipeline adalah layanan yang mengotomatisasi pergerakan dan transformasi data. Anda dapat menggunakannya untuk menjadwalkan memindahkan data input ke Amazon S3 dan menjadwalkan peluncuran klaster untuk memproses data tersebut. Sebagai contoh, pertimbangkan kasus ketika Anda memiliki server web yang merekam log lalu lintas. Jika Anda ingin menjalankan klaster mingguan untuk menganalisis data lalu lintas, Anda dapat menggunakan AWS Data Pipeline untuk menjadwalkan klaster tersebut. AWS Data Pipeline adalah alur kerja yang didorong data, sehingga satu tugas (meluncurkan klaster) dapat bergantung pada tugas lain (memindahkan data input ke Amazon S3). Ini juga memiliki fungsi coba lagi yang tangguh.

Untuk informasi selengkapnya tentang AWS Data Pipeline, lihat AWS Data Pipeline Panduan Developer, terutama tutorial mengenai Amazon EMR:

- Tutorial: Luncurkan alur kerja EMR Amazon
- Memulai: Memproses log web denganAWS Data Pipeline, Amazon EMR, dan Hive
- Tutorial: Amazon DynamoDB impor dan ekspor menggunakan AWS Data Pipeline

Memecahkan masalah cluster

Kluster EMR berjalan dalam ekosistem kompleks yang terdiri dari perangkat lunak sumber terbuka, kode aplikasi khusus, dan. Layanan AWS Ketika masalah terjadi dengan salah satu bagian ini, cluster mungkin gagal atau memakan waktu lebih lama dari yang Anda harapkan untuk menyelesaikannya. Topik berikut dapat membantu Anda mengidentifikasi masalah klaster dan cara memperbaikinya.

Topik

Ketika Anda mengembangkan aplikasi Hadoop baru, kami sarankan Anda mengaktifkan debugging dan memproses subset kecil tapi mewakili dari data Anda untuk menguji aplikasi. Anda mungkin juga ingin menjalankan aplikasi step-by-step untuk menguji setiap langkah secara terpisah. Untuk informasi selengkapnya, lihat Konfigurasi pencatatan log dan debugging klaster dan Langkah 5: Uji klaster langkah demi langkah.

Alat apa yang tersedia untuk pemecahan masalah?

Untuk mengidentifikasi dan memperbaiki kesalahan klaster, Anda dapat menggunakan alat yang dijelaskan di halaman ini. Anda mungkin perlu menginisialisasi beberapa alat saat meluncurkan cluster. Alat lain tersedia untuk setiap cluster secara default.

Topik

Lihat detail klaster EMR

Anda dapat menggunakan AWS Management Console, AWS CLI, atau API EMR untuk mengambil informasi detail tentang klaster EMR dan eksekusi pekerjaan. Untuk informasi selengkapnya tentang menggunakan AWS Management Console dan AWS CLI, lihat Melihat status dan detail klaster.

Panel detail konsol Amazon EMR

Dalam daftar Clusters di konsol EMR Amazon, Anda dapat melihat informasi tingkat tinggi tentang status setiap cluster di akun Anda dan. Wilayah AWS Daftar ini menampilkan semua cluster aktif dan dihentikan yang Anda luncurkan dalam dua bulan terakhir. Dari daftar Klaster, Anda dapat memilih sebuah Nama klaster untuk melihat detail klaster. Informasi ini tersusun dalam kategori yang berbeda untuk memudahkan navigasi.

Antarmuka pengguna Aplikasi yang tersedia di halaman detail cluster dapat berguna untuk memecahkan masalah cluster. Ini memberikan status aplikasi YARN, dan untuk beberapa, seperti aplikasi Spark Anda dapat mengebor metrik dan aspek yang berbeda seperti pekerjaan, tahapan, dan pelaksana. Untuk informasi selengkapnya, lihat Melihat riwayat aplikasi. Fitur ini hanya tersedia untuk Amazon EMR rilis 5.8.0 dan lebih tinggi.

Antarmuka baris perintah Amazon EMR

Anda dapat menemukan detail tentang cluster dari AWS CLI dengan `--describe` argumen.

API Amazon EMR

Anda dapat menemukan detail tentang sebuah klaster dari API menggunakan tindakan `DescribeJobFlows`.

Lihat detail kesalahan klaster EMR

Ketika kluster EMR berakhir dengan kesalahan, `ListClusters` API `DescribeCluster` dan mengembalikan kode kesalahan dan pesan kesalahan. Untuk kesalahan klaster tertentu, larik `ErrorDetail` data dapat membantu Anda memecahkan masalah kegagalan.

Untuk daftar kode kesalahan yang menyertakan `ErrorDetail` data, lihatKode kesalahan dengan ErrorDetail informasi.

> **Note**
>
> Kami terus menyempurnakan pesan kesalahan kami sehingga Anda menerima informasi terbaru dan relevan. Kami tidak menyarankan Anda mengurai teks dari `ErrorMessage` karena teks ini dapat berubah.

Jalankan skrip dan konfigurasikan proses EMR Amazon

Sebagai bagian dari proses pemecahan masalah, Anda mungkin merasa terbantu untuk menjalankan skrip kustom di klaster Anda atau melihat dan mengonfigurasi proses klaster.

Lihat dan mulai ulang proses aplikasi

Akan sangat membantu untuk melihat proses yang berjalan di cluster Anda untuk mendiagnosis potensi masalah. Anda dapat menghentikan dan memulai ulang proses cluster dengan menghubungkan ke node master cluster Anda. Untuk informasi selengkapnya, lihat Lihat dan mulai ulang EMR Amazon dan proses aplikasi (daemon).

Jalankan perintah dan skrip tanpa koneksi SSH

Untuk menjalankan perintah atau skrip di cluster Anda sebagai langkah, Anda dapat menggunakan `command-runner.jar` atau `script-runner.jar` alat tanpa membuat koneksi SSH ke node master. Untuk informasi selengkapnya, lihat Menjalankan perintah dan skrip di klaster EMR Amazon.

Melihat berkas log

Amazon EMR dan Hadoop sama-sama menghasilkan berkas log selama klaster berjalan. Anda dapat mengakses file log ini dari beberapa alat yang berbeda, tergantung pada konfigurasi yang Anda tentukan saat Anda meluncurkan cluster. Untuk informasi selengkapnya, lihat Konfigurasi pencatatan log dan debugging klaster.

Berkas log pada simpul utama

Setiap klaster menerbitkan berkas log ke direktori /mnt/var/log/ pada simpul utama. Berkas log ini hanya tersedia saat klaster berjalan.

Berkas log yang diarsipkan ke Amazon S3

Jika Anda meluncurkan klaster dan menentukan jalur log Amazon S3, klaster menyalin berkas log yang disimpan dalam/mnt/var/log/ pada simpul utama untuk Amazon S3 dalam interval 5 menit. Hal ini memastikan bahwa Anda memiliki akses ke file berkas log bahkan setelah klaster diakhiri. Karena file diarsipkan dalam interval 5 menit, beberapa menit terakhir dari klaster yang tiba-tiba diakhiri mungkin tidak tersedia.

Pantau kinerja cluster EMR

Amazon EMR menyediakan beberapa alat untuk memantau performa klaster Anda.

Antarmuka web Hadoop

Setiap klaster menerbitkan satu set antarmuka web pada simpul utama yang berisi informasi tentang klaster. Anda dapat mengakses halaman web ini dengan menggunakan terowongan SSH untuk menghubungkan mereka pada simpul utama. Untuk informasi selengkapnya, lihat Melihat antarmuka web yang di-host pada klaster Amazon EMR.

CloudWatch metrik

Setiap cluster melaporkan metrik ke CloudWatch. CloudWatch adalah layanan web yang melacak metrik, dan yang dapat Anda gunakan untuk mengatur alarm pada metrik tersebut. Untuk informasi selengkapnya, lihat Memantau metrik Amazon EMR dengan CloudWatch.

Lihat dan mulai ulang EMR Amazon dan proses aplikasi (daemon)

Ketika Anda memecahkan masalah klaster, Anda mungkin ingin mencantumkan proses yang berjalan. Anda mungkin juga ingin menghentikan atau memulai ulang ProcesSess. Misalnya, Anda dapat memulai ulang proses setelah mengubah konfigurasi atau melihat masalah dengan proses tertentu setelah Anda menganalisis file log dan pesan kesalahan.

Ada dua jenis proses yang berjalan di cluster: Amazon EMR proses (misalnya, instance-controller dan Log Pusher), dan proses yang terkait dengan aplikasi yang diinstal pada cluster (misalnya,, dan). hadoop-hdfs-namenode hadoop-yarn-resourcemanager

Untuk bekerja dengan proses langsung pada cluster, Anda harus terlebih dahulu terhubung ke node master. Untuk informasi selengkapnya, lihat Connect ke sebuah cluster.

Melihat proses yang berjalan

Metode yang Anda gunakan untuk melihat proses yang sedang berjalan di klaster berbeda sesuai dengan versi EMR Amazon yang Anda gunakan.

EMR 5.30.0 and 6.0.0 and later

Example : Daftar semua proses yang berjalan

Contoh berikut menggunakan `systemctl` dan menentukan `--type` untuk melihat semua proses.

```
systemctl --type=service
```

Example : Daftar proses spesifik

Contoh berikut mencantumkan semua proses dengan nama yang berisi`hadoop`.

```
systemctl --type=service | grep -i hadoop
```

Contoh output:

```
 hadoop-hdfs-namenode.service           loaded active running Hadoop namenode
 hadoop-httpfs.service                  loaded active running Hadoop httpfs
 hadoop-kms.service                     loaded active running Hadoop kms
 hadoop-mapreduce-historyserver.service loaded active running Hadoop historyserver
 hadoop-state-pusher.service            loaded active running Daemon process that
 processes and serves EMR metrics data.
 hadoop-yarn-proxyserver.service        loaded active running Hadoop proxyserver
 hadoop-yarn-resourcemanager.service    loaded active running Hadoop resourcemanager
 hadoop-yarn-timelineserver.service     loaded active running Hadoop timelineserver
```

Example : Lihat laporan status terperinci untuk proses tertentu

Contoh berikut menampilkan laporan status rinci untuk `hadoop-hdfs-namenode` layanan.

```
sudo systemctl status hadoop-hdfs-namenode
```

Contoh output:

```
hadoop-hdfs-namenode.service - Hadoop namenode
```

```
   Loaded: loaded (/etc/systemd/system/hadoop-hdfs-namenode.service; enabled; vendor
 preset: disabled)
   Active: active (running) since Wed 2021-08-18 21:01:46 UTC; 26min ago
 Main PID: 9733 (java)
    Tasks: 0
   Memory: 1.1M
   CGroup: /system.slice/hadoop-hdfs-namenode.service
           # 9733 /etc/alternatives/jre/bin/java -Dproc_namenode -Xmx1843m -server -
XX:OnOutOfMemoryError=kill -9 %p ...

Aug 18 21:01:37 ip-172-31-20-123 systemd[1]: Starting Hadoop namenode...
Aug 18 21:01:37 ip-172-31-20-123 su[9715]: (to hdfs) root on none
Aug 18 21:01:37 ip-172-31-20-123 hadoop-hdfs-namenode[9683]: starting namenode,
 logging to /var/log/hadoop-hdfs/ha...out
Aug 18 21:01:46 ip-172-31-20-123 hadoop-hdfs-namenode[9683]: Started Hadoop
 namenode:[  OK  ]
Aug 18 21:01:46 ip-172-31-20-123 systemd[1]: Started Hadoop namenode.
Hint: Some lines were ellipsized, use -l to show in full.
```

EMR 4.x - 5.29.0

Example : Daftar semua proses yang berjalan

Contoh berikut mencantumkan semua proses yang berjalan.

```
initctl list
```

EMR 2.x - 3.x

Example : Daftar semua proses yang berjalan

Contoh berikut mencantumkan semua proses yang berjalan.

```
ls /etc/init.d/
```

Menghentikan dan memulai kembali proses

Setelah Anda menentukan proses apa yang sedang berjalan, Anda dapat menghentikannya lalu memulai ulang jika diperlukan.

EMR 5.30.0 and 6.0.0 and later

Example : Hentikan proses

Contoh berikut menghentikan `hadoop-hdfs-namenode` proses.

```
sudo systemctl stop hadoop-hdfs-namenode
```

Anda dapat meminta `status` untuk memverifikasi bahwa proses dihentikan.

```
sudo systemctl status hadoop-hdfs-namenode
```

Contoh output:

```
hadoop-hdfs-namenode.service - Hadoop namenode
  Loaded: loaded (/etc/systemd/system/hadoop-hdfs-namenode.service; enabled; vendor
 preset: disabled)
  Active: failed (Result: exit-code) since Wed 2021-08-18 21:37:50 UTC; 8s ago
Main PID: 9733 (code=exited, status=143)
```

Example : Memulai proses

Contoh berikut memulai `hadoop-hdfs-namenode` proses.

```
sudo systemctl start hadoop-hdfs-namenode
```

Anda dapat menanyakan status untuk memverifikasi bahwa proses sedang berjalan.

```
sudo systemctl status hadoop-hdfs-namenode
```

Contoh output:

```
hadoop-hdfs-namenode.service - Hadoop namenode
   Loaded: loaded (/etc/systemd/system/hadoop-hdfs-namenode.service; enabled; vendor
 preset: disabled)
   Active: active (running) since Wed 2021-08-18 21:38:24 UTC; 2s ago
  Process: 13748 ExecStart=/etc/init.d/hadoop-hdfs-namenode start (code=exited,
 status=0/SUCCESS)
 Main PID: 13800 (java)
    Tasks: 0
   Memory: 1.1M
```

```
  CGroup: /system.slice/hadoop-hdfs-namenode.service
          # 13800 /etc/alternatives/jre/bin/java -Dproc_namenode -Xmx1843m -server
 -XX:OnOutOfMemoryError=kill -9 %p...
```

EMR 4.x - 5.29.0

Example : Hentikan proses yang sedang berjalan

Contoh berikut menghentikan `hadoop-hdfs-namenode` layanan.

```
sudo stop hadoop-hdfs-namenode
```

Example : Mulai ulang proses yang dihentikan

Contoh berikut memulai ulang `hadoop-hdfs-namenode` layanan. Anda harus menggunakan `start` perintah dan tidak`restart`.

```
sudo start hadoop-hdfs-namenode
```

Example : Periksa status proses

Berikut ini mengambil status untuk`hadoop-hdfs-namenode`. Anda dapat menggunakan `status` perintah untuk memverifikasi bahwa proses telah berhenti atau dimulai.

```
sudo status hadoop-hdfs-namenode
```

EMR 2.x - 3.x

Example : Hentikan proses aplikasi

Contoh berikut menghentikan `hadoop-hdfs-namenode` layanan, yang dikaitkan dengan versi Amazon EMR yang diinstal pada cluster.

```
sudo /etc/init.d/hadoop-hdfs-namenode stop
```

Example : Mulai ulang proses aplikasi

Contoh perintah berikut memulai ulang `hadoop-hdfs-namenode` proses:

```
sudo /etc/init.d/hadoop-hdfs-namenode start
```

Example : Hentikan proses EMR Amazon

Contoh berikut menghentikan proses, seperti instance-controller, yang tidak terkait dengan versi Amazon EMR di cluster.

```
sudo /sbin/stop instance-controller
```

Example : Mulai ulang proses EMR Amazon

Contoh berikut memulai ulang proses, seperti instance-controller, yang tidak terkait dengan versi Amazon EMR di cluster.

```
sudo /sbin/start instance-controller
```

> **Note**
>
> Perintah `/sbin/start, stop` dan `restart` adalah symlink ke `/sbin/intictl`. Untuk informasi selengkapnya tentang `initctl`, lihat halaman man initctl dengan mengetik `man initctl` pada prompt perintah.

Kesalahan umum di Amazon EMR

Terkadang, cluster gagal atau lambat memproses data. Bagian berikut mencantumkan beberapa masalah klaster umum Dengan saran tentang cara memperbaikinya.

Topik

- Kode kesalahan dengan ErrorDetail informasi
- Kesalahan sumber daya
- Kesalahan input dan output
- Kesalahan izin
- Kesalahan Klaster Hive
- Kesalahan VPC
- Kesalahan klaster streaming
- Kesalahan klaster JAR kustom
- AWS GovCloud Kesalahan (AS-Barat)

- Temukan cluster yang hilang

Kode kesalahan dengan ErrorDetail informasi

Ketika kluster EMR berakhir dengan kesalahan, `ListClusters` API `DescribeCluster` dan mengembalikan kode kesalahan dan pesan kesalahan. Untuk beberapa kesalahan cluster, array `ErrorDetail` data dapat membantu Anda memecahkan masalah kegagalan.

Kesalahan yang menyertakan `ErrorDetail` array memberikan rincian berikut:

`ErrorCode`

Kode kesalahan unik yang dapat Anda gunakan untuk akses terprogram.

`ErrorData`

Daftar pengidentifikasi dalam pasangan nilai kunci yang dapat Anda gunakan untuk pemrograman atau pencarian manual. Untuk deskripsi `ErrorData` nilai yang disertakan kode kesalahan, lihat halaman pemecahan masalah untuk kode kesalahan.

`ErrorMessage`

Deskripsi kesalahan dengan tautan ke informasi lebih lanjut dalam dokumentasi EMR Amazon.

> **Note**
>
> Kami tidak menyarankan Anda mengurai teks dari `ErrorMessage` karena teks ini dapat berubah.

Kode kesalahan berdasarkan kategori

- Kode kesalahan kegagalan bootstrap
- Kode kesalahan internal
- Kode kesalahan kegagalan validasi

Kode kesalahan kegagalan bootstrap

Bagian berikut memberikan informasi pemecahan masalah untuk kode kesalahan kegagalan bootstrap.

Topik

BOOTSTRAP_FAILURE_PRIMARY_WITH_NON_ZERO_CODE

Gambaran Umum

Ketika sebuah cluster berakhir dengan `BOOTSTRAP_FAILURE_PRIMARY_WITH_NON_ZERO_CODE` kesalahan, tindakan bootstrap telah gagal dalam contoh utama. Untuk informasi selengkapnya tentang tindakan bootstrap, lihatBuat tindakan bootstrap untuk menginstal perangkat lunak tambahan.

Resolusi

Untuk mengatasi kesalahan ini, tinjau detail yang dikembalikan dalam kesalahan API, modifikasi skrip tindakan bootstrap Anda, dan buat cluster baru dengan tindakan bootstrap yang diperbarui.

Untuk memecahkan masalah klaster EMR yang gagal, lihat `ErrorDetail` informasi yang dikembalikan dari dan API. `DescribeCluster` `ListClusters` Untuk informasi selengkapnya, lihat Kode kesalahan dengan ErrorDetail informasi. `ErrorData`Array dalam `ErrorDetail` mengembalikan informasi berikut untuk kode kesalahan ini:

`primary-instance-id`

ID dari instance utama di mana tindakan bootstrap gagal.

`bootstrap-action`

Nomor urut untuk tindakan bootstrap yang gagal. Skrip dengan `bootstrap-action` nilai `1` adalah tindakan bootstrap pertama yang dijalankan pada instance.

`return-code`

Kode pengembalian untuk tindakan bootstrap yang gagal.

`amazon-s3-path`

Lokasi Amazon S3 dari tindakan bootstrap yang gagal.

`public-doc`

URL publik dokumentasi untuk kode kesalahan.

Langkah-langkah untuk menyelesaikan

Lakukan langkah-langkah berikut untuk mengidentifikasi dan memperbaiki akar penyebab kesalahan tindakan bootstrap. Kemudian luncurkan cluster baru.

1. Tinjau file log tindakan bootstrap di Amazon S3 untuk mengidentifikasi akar penyebab kegagalan tersebut. Untuk mempelajari lebih lanjut tentang cara melihat log EMR Amazon, lihat. Melihat berkas log
2. Jika Anda mengaktifkan log klaster saat membuat instance, lihat `stdout` log untuk informasi selengkapnya. Anda dapat menemukan `stdout` log untuk tindakan bootstrap di lokasi Amazon S3 ini:

   ```
   s3://EXAMPLE-BUCKET/logs/Your_Cluster_Id/node/Primary_Instance_Id/bootstrap-actions/Failed_Bootstrap_Action_Number/stdout.gz
   ```

   Untuk informasi selengkapnya tentang log klaster, lihatKonfigurasi pencatatan log dan debugging klaster.
3. Untuk menentukan kegagalan tindakan bootstrap, tinjau pengecualian di `stdout` log, dan `return-code` nilainya. `ErrorData`
4. Gunakan temuan Anda dari langkah sebelumnya untuk merevisi tindakan bootstrap Anda sehingga menghindari pengecualian atau dapat menangani pengecualian dengan anggun saat terjadi.
5. Luncurkan cluster baru dengan tindakan bootstrap Anda yang diperbarui.

BOOTSTRAP_FAILURE_BA_DOWNLOAD_FAILED_PRIMARY

Gambaran Umum

Kluster berakhir dengan `BOOTSTRAP_FAILURE_BA_DOWNLOAD_FAILED_PRIMARY` kesalahan saat instance utama tidak dapat mengunduh skrip tindakan bootstrap dari lokasi Amazon S3 yang Anda tentukan. Penyebab potensial meliputi:

- File skrip tindakan bootstrap tidak berada di lokasi Amazon S3 yang ditentukan.
- Peran layanan untuk instans Amazon EC2 di cluster (juga disebut profil instans EC2 untuk Amazon EMR) tidak memiliki izin untuk mengakses bucket Amazon S3 tempat skrip tindakan bootstrap berada. Untuk informasi selengkapnya tentang peran layanan, lihat Peran layanan untuk instans EC2 klaster (profil instans EC2).

Untuk informasi selengkapnya tentang tindakan bootstrap, lihatBuat tindakan bootstrap untuk menginstal perangkat lunak tambahan.

Resolusi

Untuk mengatasi kesalahan ini, pastikan bahwa instance utama Anda memiliki akses yang sesuai ke skrip tindakan bootstrap.

Untuk memecahkan masalah klaster EMR yang gagal, lihat `ErrorDetail` informasi yang dikembalikan dari dan API. `DescribeCluster` `ListClusters` Untuk informasi selengkapnya, lihat Kode kesalahan dengan ErrorDetail informasi. `ErrorData`Array dalam `ErrorDetail` mengembalikan informasi berikut untuk kode kesalahan ini:

`primary-instance-id`

ID dari instance utama di mana tindakan bootstrap gagal.

`bootstrap-action`

Nomor urut untuk tindakan bootstrap yang gagal. Skrip dengan `bootstrap-action` nilai 1 adalah tindakan bootstrap pertama yang dijalankan pada instance.

`amazon-s3-path`

Lokasi Amazon S3 dari tindakan bootstrap yang gagal.

`public-doc`

URL publik dokumentasi untuk kode kesalahan.

Langkah-langkah untuk menyelesaikan

Lakukan langkah-langkah berikut untuk mengidentifikasi dan memperbaiki akar penyebab kesalahan tindakan bootstrap. Kemudian luncurkan cluster baru.

Langkah pemecahan masalah

1. Gunakan `amazon-s3-path` nilai dari `ErrorData` array untuk menemukan skrip tindakan bootstrap yang relevan di Amazon S3.
2. Jika Anda mengaktifkan log klaster saat membuat instance, lihat `stdout` log untuk informasi selengkapnya. Anda dapat menemukan `stdout` log untuk tindakan bootstrap di lokasi Amazon S3 ini:

```
s3://EXAMPLE-BUCKET/logs/Your_Cluster_Id/node/Primary_Instance_Id/bootstrap-
actions/Failed_Bootstrap_Action_Number/stdout.gz
```

Untuk informasi selengkapnya tentang log klaster, lihatKonfigurasi pencatatan log dan debugging klaster.

3. Untuk menentukan kegagalan tindakan bootstrap, tinjau pengecualian di `stdout` log, dan `return-code` nilainya. `ErrorData`
4. Gunakan temuan Anda dari langkah sebelumnya untuk merevisi tindakan bootstrap Anda sehingga menghindari pengecualian atau dapat menangani pengecualian dengan anggun saat terjadi.
5. Luncurkan cluster baru dengan tindakan bootstrap Anda yang diperbarui.

BOOTSTRAP_FAILURE_FILE_NOT_FOUND_PRIMARY

Gambaran Umum

`BOOTSTRAP_FAILURE_FILE_NOT_FOUND_PRIMARY`Kesalahan menunjukkan bahwa instance utama tidak dapat menemukan skrip tindakan bootstrap yang baru saja diunduh instance dari bucket Amazon S3 yang ditentukan.

Resolusi

Untuk mengatasi kesalahan ini, konfirmasikan bahwa instance utama Anda memiliki akses yang sesuai ke skrip tindakan bootstrap.

Untuk memecahkan masalah klaster EMR yang gagal, lihat `ErrorDetail` informasi yang dikembalikan dari dan API. `DescribeCluster` `ListClusters` Untuk informasi selengkapnya, lihat Kode kesalahan dengan ErrorDetail informasi. `ErrorData`Array dalam `ErrorDetail` mengembalikan informasi berikut untuk kode kesalahan ini:

`primary-instance-id`

ID dari instance utama di mana tindakan bootstrap gagal.

`bootstrap-action`

Nomor urut untuk tindakan bootstrap yang gagal. Skrip dengan `bootstrap-action` nilai 1 adalah tindakan bootstrap pertama yang dijalankan pada instance.

`amazon-s3-path`

Lokasi Amazon S3 dari tindakan bootstrap yang gagal.

`public-doc`

URL publik dokumentasi untuk kode kesalahan.

Langkah-langkah untuk menyelesaikan

Lakukan langkah-langkah berikut untuk mengidentifikasi dan memperbaiki akar penyebab kesalahan tindakan bootstrap. Kemudian luncurkan cluster baru.

1. Untuk menemukan skrip tindakan bootstrap yang relevan di Amazon S3, gunakan `amazon-s3-path` nilai dari array. `ErrorData`
2. Tinjau file log tindakan bootstrap di Amazon S3 untuk mengidentifikasi akar penyebab kegagalan tersebut. Untuk mempelajari lebih lanjut tentang cara melihat log EMR Amazon, lihat. Melihat berkas log

   > **Note**
   >
   > Jika Anda tidak mengaktifkan log untuk cluster Anda, Anda harus membuat cluster baru dengan konfigurasi dan tindakan bootstrap yang sama. Untuk memastikan log cluster diaktifkan, lihatKonfigurasi pencatatan log dan debugging klaster.

3. Tinjau `stdout` log untuk tindakan bootstrap Anda dan konfirmasikan bahwa tidak ada proses khusus yang menghapus file di `/emr/instance-controller/lib/bootstrap-actions` folder pada instance utama Anda. Anda dapat menemukan `stdout` log untuk tindakan bootstrap di lokasi Amazon S3 ini:

   ```
   s3://EXAMPLE-BUCKET/logs/Your_Cluster_Id/node/Primary_Instance_Id/bootstrap-actions/Failed_Bootstrap_Action_Number/stdout.gz
   ```

4. Luncurkan cluster baru dengan tindakan bootstrap Anda yang diperbarui.

Kode kesalahan internal

Bagian berikut menyediakan informasi pemecahan masalah untuk kode kesalahan internal.

Topik

- INTERNAL_ERROR_EC2_INSUFFICIENT_CAPACITY_AZ
- INTERNAL_ERROR_SPOT_PRICE_INCREASE_PRIMARY
- INTERNAL_ERROR_SPOT_NO_CAPACITY_PRIMARY

INTERNAL_ERROR_EC2_INSUFFICIENT_CAPACITY_AZ

Gambaran Umum

Cluster diakhiri dengan `INTERNAL_ERROR_EC2_INSUFFICIENT_CAPACITY_AZ` kesalahan saat Availability Zone yang dipilih tidak memiliki kapasitas yang cukup untuk memenuhi permintaan jenis instans Amazon EC2 Anda. Subnet yang Anda pilih untuk sebuah klaster menentukan Availability Zone. Untuk informasi selengkapnya tentang subnet untuk Amazon EMR, lihat. Mengkonfigurasi jaringan

Resolusi

Untuk mengatasi kesalahan ini, ubah konfigurasi tipe instans Anda dan buat klaster baru dengan permintaan yang diperbarui.

Untuk memecahkan masalah klaster EMR yang gagal, lihat `ErrorDetail` informasi yang dikembalikan dari dan API. `DescribeCluster` `ListClusters` Untuk informasi selengkapnya, lihat Kode kesalahan dengan ErrorDetail informasi. `ErrorData`Array dalam `ErrorDetail` mengembalikan informasi berikut untuk kode kesalahan ini:

`instance-type`

Jenis instance yang di luar kapasitas.

`availability-zone`

Availability Zone yang diselesaikan oleh subnet Anda.

`public-doc`

URL publik dokumentasi untuk kode kesalahan.

Langkah-langkah untuk menyelesaikan

Lakukan langkah-langkah berikut untuk mengidentifikasi dan memperbaiki akar penyebab kesalahan konfigurasi cluster:

- Tinjau praktik terbaik untuk konfigurasi cluster. Lihat Praktik terbaik untuk konfigurasi klaster di Panduan Manajemen EMR Amazon.
- Memecahkan masalah peluncuran dan meninjau konfigurasi Anda. Lihat Memecahkan masalah peluncuran instans di Panduan Pengguna Amazon EC2 untuk Instans Linux.
- Luncurkan cluster baru dengan konfigurasi cluster Anda yang diperbarui.

INTERNAL_ERROR_SPOT_PRICE_INCREASE_PRIMARY

Gambaran Umum

Kluster berakhir dengan `INTERNAL_ERROR_SPOT_PRICE_INCREASE_PRIMARY` kesalahan saat Amazon EMR tidak dapat memenuhi permintaan Instans Spot Anda untuk node utama karena instance tidak tersedia pada atau di bawah harga Spot maksimum Anda. Untuk informasi selengkapnya, lihat Instans Spot dalam Panduan Pengguna Amazon EC2 untuk Instans Linux.

Resolusi

Untuk mengatasi kesalahan ini, tentukan jenis instance untuk klaster Anda yang berada dalam target harga Anda, atau tingkatkan batas harga Anda untuk jenis instans yang sama.

Untuk memecahkan masalah klaster EMR yang gagal, lihat `ErrorDetail` informasi yang dikembalikan dari dan API. `DescribeCluster` `ListClusters` Untuk informasi selengkapnya, lihat Kode kesalahan dengan ErrorDetail informasi. `ErrorData`Array dalam `ErrorDetail` mengembalikan informasi berikut untuk kode kesalahan ini:

`primary-instance-id`

ID untuk instance utama cluster yang gagal.

`instance-type`

Jenis instance yang di luar kapasitas.

`availability-zone`

Availability Zone tempat subnet Anda berada.

`public-doc`

URL publik dokumentasi untuk kode kesalahan.

Langkah-langkah untuk menyelesaikan

Lakukan langkah-langkah berikut untuk memecahkan masalah strategi konfigurasi klaster Anda, lalu luncurkan klaster baru:

1. Tinjau praktik terbaik untuk Instans Spot Amazon EC2 dan tinjau strategi konfigurasi klaster Anda. Untuk informasi selengkapnya, lihat Praktik terbaik untuk EC2 Spot di Panduan Pengguna Amazon EC2 untuk Instans Linux dan. Praktik terbaik untuk konfigurasi klaster
2. Ubah konfigurasi tipe instans atau Availability Zone Anda dan buat klaster baru dengan permintaan yang diperbarui.
3. Jika masalah berlanjut, gunakan kapasitas On-Demand untuk instans utama Anda.

INTERNAL_ERROR_SPOT_NO_CAPACITY_PRIMARY

Gambaran Umum

Sebuah klaster berakhir dengan `INTERNAL_ERROR_SPOT_NO_CAPACITY_PRIMARY` kesalahan ketika tidak ada kapasitas yang cukup untuk memenuhi permintaan Instans Spot untuk node utama Anda. Untuk informasi selengkapnya, lihat Instans Spot dalam Panduan Pengguna Amazon EC2 untuk Instans Linux.

Resolusi

Untuk mengatasi kesalahan ini, tentukan jenis instance untuk klaster Anda yang berada dalam target harga Anda, atau tingkatkan batas harga Anda untuk jenis instans yang sama.

Untuk memecahkan masalah klaster EMR yang gagal, lihat `ErrorDetail` informasi yang dikembalikan dari dan API. `DescribeCluster` `ListClusters` Untuk informasi selengkapnya, lihat Kode kesalahan dengan ErrorDetail informasi. `ErrorData`Array dalam `ErrorDetail` mengembalikan informasi berikut untuk kode kesalahan ini:

`primary-instance-id`

ID untuk instance utama cluster yang gagal.

`instance-type`

Jenis instance yang di luar kapasitas.

`availability-zone`

Availability Zone yang diselesaikan oleh subnet Anda.

`public-doc`

URL publik dokumentasi untuk kode kesalahan.

Langkah-langkah untuk menyelesaikan

Lakukan langkah-langkah berikut untuk memecahkan masalah strategi konfigurasi klaster Anda, lalu luncurkan klaster baru:

1. Tinjau praktik terbaik untuk Instans Spot Amazon EC2 dan tinjau strategi konfigurasi klaster Anda. Untuk informasi selengkapnya, lihat Praktik terbaik untuk EC2 Spot di Panduan Pengguna Amazon EC2 untuk Instans Linux dan. Praktik terbaik untuk konfigurasi klaster
2. Ubah konfigurasi tipe instans Anda dan buat klaster baru dengan permintaan yang diperbarui.
3. Jika masalah berlanjut, gunakan kapasitas On-Demand untuk instans utama Anda.

Kode kesalahan kegagalan validasi

Bagian berikut menyediakan informasi pemecahan masalah untuk kode kesalahan kegagalan validasi.

Topik

- VALIDATION_ERROR_SUBNET_NOT_FROM_ONE_VPC
- VALIDATION_ERROR_SECURITY_GROUP_NOT_FROM_ONE_VPC
- VALIDATION_ERROR_INVALID_SSH_KEY_NAME
- VALIDATION_ERROR_INSTANCE_TYPE_NOT_SUPPORTED

VALIDATION_ERROR_SUBNET_NOT_FROM_ONE_VPC

Gambaran Umum

Ketika cluster Anda dan subnet yang Anda referensikan untuk cluster Anda milik virtual private cloud (VPC) yang berbeda, cluster berakhir dengan kesalahan. `VALIDATION_ERROR_SUBNET_NOT_FROM_ONE_VPC` Anda dapat meluncurkan cluster dengan Amazon EMR dengan konfigurasi armada instance di seluruh subnet dalam VPC. Untuk informasi selengkapnya tentang armada instans, lihat Mengkonfigurasi armada instans di Panduan Manajemen EMR Amazon.

Resolusi

Untuk mengatasi kesalahan ini, gunakan subnet yang termasuk dalam VPC yang sama dengan cluster.

Untuk memecahkan masalah klaster EMR yang gagal, lihat `ErrorDetail` informasi yang dikembalikan dari dan API. `DescribeCluster` `ListClusters` Untuk informasi selengkapnya, lihat Kode kesalahan dengan ErrorDetail informasi. `ErrorData`Array dalam `ErrorDetail` mengembalikan informasi berikut untuk kode kesalahan ini:

`vpc`

Untuk setiap pasangan subnet:VPC, ID untuk VPC yang dimiliki subnet.

`subnet`

Untuk setiap pasangan subnet:VPC, ID untuk subnet.

`public-doc`

URL publik dokumentasi untuk kode kesalahan.

Langkah-langkah untuk menyelesaikan

Lakukan langkah-langkah berikut untuk mengidentifikasi dan memperbaiki kesalahan:

1. Tinjau ID subnet yang tercantum dalam `ErrorData` array dan konfirmasikan bahwa ID tersebut milik VPC tempat Anda ingin meluncurkan cluster EMR.
2. Ubah konfigurasi subnet Anda. Anda dapat menggunakan salah satu metode berikut untuk menemukan semua subnet publik dan pribadi yang tersedia di VPC.

   - Arahkan ke Konsol VPC Amazon. Pilih Subnet dan daftar semua subnet yang berada di dalam Wilayah AWS untuk cluster Anda. Untuk hanya menemukan subnet publik atau pribadi, terapkan filter alamat IPv4 publik yang ditetapkan otomatis. Untuk menemukan dan memilih subnet di VPC yang digunakan cluster Anda, gunakan opsi Filter by VPC. Untuk informasi selengkapnya tentang cara membuat subnet, lihat Membuat subnet di Panduan Pengguna Amazon Virtual Private Cloud.
   - Gunakan AWS CLI untuk menemukan semua subnet publik dan pribadi yang tersedia di VPC yang digunakan cluster Anda. Untuk informasi selengkapnya, lihat API deskripsi-subnet. Untuk membuat subnet baru di VPC, lihat API create-subnet.
3. Luncurkan cluster baru dengan subnet dari VPC yang sama dengan cluster.

VALIDATION_ERROR_SECURITY_GROUP_NOT_FROM_ONE_VPC

Gambaran Umum

Ketika klaster dan grup keamanan yang Anda tetapkan ke cluster Anda termasuk dalam virtual private cloud (VPC) yang berbeda, cluster berakhir dengan kesalahan. `VALIDATION_ERROR_SECURITY_GROUP_NOT_FROM_ONE_VPC` Untuk informasi selengkapnya tentang grup keamanan, lihat Menentukan grup keamanan terkelola Amazon EMR dan grup keamanan tambahan danMengendalikan lalu lintas jaringan dengan grup keamanan.

Resolusi

Untuk mengatasi kesalahan ini, gunakan grup keamanan yang termasuk dalam VPC yang sama dengan cluster.

Untuk memecahkan masalah klaster EMR yang gagal, lihat `ErrorDetail` informasi yang dikembalikan dari dan API. `DescribeCluster` `ListClusters` Untuk informasi selengkapnya, lihat Kode kesalahan dengan ErrorDetail informasi. `ErrorData`Array dalam `ErrorDetail` mengembalikan informasi berikut untuk kode kesalahan ini:

`vpc`

Untuk setiap pasangan Security-group:VPC, ID untuk VPC yang menjadi milik grup keamanan.

`security-group`

Untuk setiap pasangan Security-group:VPC, ID untuk grup keamanan.

`public-doc`

URL publik dokumentasi untuk kode kesalahan.

Langkah-langkah untuk menyelesaikan

Lakukan langkah-langkah berikut untuk mengidentifikasi dan memperbaiki kesalahan:

1. Tinjau ID grup keamanan yang tercantum dalam `ErrorData` larik dan konfirmasikan bahwa ID tersebut milik VPC tempat Anda ingin meluncurkan cluster EMR.
2. Arahkan ke Konsol VPC Amazon. Pilih Grup keamanan untuk mencantumkan semua grup keamanan dalam Wilayah yang Anda pilih. Temukan grup keamanan dari VPC yang sama dengan cluster Anda, lalu ubah konfigurasi grup keamanan Anda.

3. Luncurkan cluster baru dengan grup keamanan dari VPC yang sama dengan cluster.

VALIDATION_ERROR_INVALID_SSH_KEY_NAME

Gambaran Umum

Kluster akan berakhir dengan `VALIDATION_ERROR_INVALID_SSH_KEY_NAME` error saat Anda menggunakan key pair Amazon EC2 yang tidak valid untuk SSH ke instans utama. Nama key pair mungkin salah, atau key pair mungkin tidak ada di requestWilayah AWS. Untuk informasi selengkapnya tentang pasangan kunci, lihat pasangan kunci Amazon EC2 dan instans Linux di Panduan Pengguna Amazon EC2 untuk Instans Linux.

Resolusi

Untuk mengatasi kesalahan ini, buat cluster baru dengan nama key pair SSH yang valid.

Untuk memecahkan masalah klaster EMR yang gagal, lihat `ErrorDetail` informasi yang dikembalikan dari dan API. `DescribeCluster` `ListClusters` Untuk informasi selengkapnya, lihat Kode kesalahan dengan ErrorDetail informasi. `ErrorDataArray` dalam `ErrorDetail` mengembalikan informasi berikut untuk kode kesalahan ini:

`ssh-key`

Nama key pair SSH yang Anda berikan saat Anda membuat cluster.

`public-doc`

URL publik dokumentasi untuk kode kesalahan.

Langkah-langkah untuk menyelesaikan

Lakukan langkah-langkah berikut untuk mengidentifikasi dan memperbaiki kesalahan:

1. Periksa file.pem *`keypair`* Anda dan konfirmasikan bahwa itu cocok dengan nama kunci SSH yang Anda lihat di konsol EMR Amazon.
2. Arahkan ke konsol Amazon EC2. Verifikasi bahwa nama kunci SSH yang Anda gunakan tersedia di Wilayah AWS klaster Anda. Anda dapat menemukan Wilayah AWS di sebelah ID akun Anda di bagian atasAWS Management Console.
3. Luncurkan cluster baru dengan nama kunci SSH yang valid.

VALIDATION_ERROR_INSTANCE_TYPE_NOT_SUPPORTED

Gambaran Umum

Kluster diakhiri dengan `VALIDATION_ERROR_INSTANCE_TYPE_NOT_SUPPORTED` kesalahan saat Wilayah AWS dan Availability Zone untuk klaster Anda tidak mendukung jenis instans yang ditentukan untuk satu atau beberapa grup instans. Amazon EMR mungkin mendukung jenis instans di satu Availability Zone dalam Wilayah tetapi tidak yang lain. Subnet yang Anda pilih untuk klaster menentukan Availability Zone dalam Region. Untuk daftar jenis instans dan Wilayah yang didukung Amazon EMR, lihat. Tipe instans yang didukung

Resolusi

Untuk mengatasi kesalahan ini, tentukan jenis instans untuk klaster yang didukung Amazon EMR di Wilayah dan Zona Ketersediaan tempat Anda meminta klaster.

Untuk memecahkan masalah klaster EMR yang gagal, lihat `ErrorDetail` informasi yang dikembalikan dari dan API. `DescribeCluster` `ListClusters` Untuk informasi selengkapnya, lihat Kode kesalahan dengan ErrorDetail informasi. `ErrorData`Array dalam `ErrorDetail` mengembalikan informasi berikut untuk kode kesalahan ini:

`instance-types`

Daftar jenis instans yang tidak didukung.

`availability-zones`

Daftar Availability Zones yang subnet Anda selesaikan.

`public-doc`

URL publik dokumentasi untuk kode kesalahan.

Langkah-langkah untuk menyelesaikan

Lakukan langkah-langkah berikut untuk mengidentifikasi dan memperbaiki kesalahan:

1. Gunakan AWS CLI untuk mengambil jenis instance yang tersedia di Availability Zone. Untuk melakukan ini, Anda dapat menggunakan `ec2 describe-instance-type-offerings` perintah untuk memfilter jenis instance yang tersedia berdasarkan lokasi (Wilayah AWSatau Availability Zone). Misalnya, perintah berikut mengembalikan jenis instance yang ditawarkan di AZ tertentu,*`us-east-2a`*.

```
aws ec2 describe-instance-type-offerings --location-type "availability-
zone" --filters Name=location,Values=us-east-2a --region us-east-2 --query
 "InstanceTypeOfferings[*].[InstanceType]" --output text | sort
```

Untuk mempelajari lebih lanjut tentang cara menemukan jenis instans yang tersedia, lihat Menemukan jenis instans Amazon EC2.

2. Setelah menentukan jenis instance yang tersedia di Region dan Availability Zone yang sama dengan cluster, pilih salah satu resolusi berikut untuk melanjutkan:

   a. Buat klaster baru, dan pilih subnet untuk klaster yang ada di Availability Zone tempat jenis instans yang dipilih tersedia dan didukung oleh Amazon EMR.

   b. Buat klaster baru di subnet Region dan Amazon EC2 yang sama dengan cluster yang gagal, tetapi dengan tipe instans yang didukung di lokasi tersebut oleh Amazon EMR.

Untuk daftar jenis instans dan Wilayah yang didukung Amazon EMR, lihat. Tipe instans yang didukung Untuk membandingkan kemampuan jenis instans, lihat jenis instans Amazon EC2.

Kesalahan sumber daya

Kesalahan berikut ini biasanya disebabkan oleh sumber daya terbatas pada klaster.

Topik

- Klaster berakhir dengan NO_SLAVE_LEFT dan simpul inti FAILED_BY_MASTER
- Tidak dapat mereplikasi blok, hanya berhasil mereplikasi ke nol simpul.
- KUOTA EC2 TERLAMPAUI
- Terlalu banyak kegagalan mengambil
- File hanya dapat direplikasi ke 0 simpul bukan 1
- Node yang terdaftar penolakan
- Kesalahan throttling
- Tipe instans tidak didukung
- EC2 berada di luar kapasitas

Klaster berakhir dengan NO_SLAVE_LEFT dan simpul inti FAILED_BY_MASTER

Biasanya, hal ini terjadi karena proteksi pengakhiran dinonaktifkan, dan semua simpul inti melebihi kapasitas penyimpanan disk yang ditentukan oleh ambang pemanfaatan maksimum di klasifikasi konfigurasi `yarn-site`, yang sesuai dengan file `yarn-site.xml`. Nilainya adalah 90% secara default. Ketika pemanfaatan disk untuk node inti melebihi ambang batas pemanfaatan, layanan NodeManager kesehatan YARN melaporkan node sebagai. `UNHEALTHY` Sementara dalam keadaan ini, Amazon EMR menolak daftar node dan tidak mengalokasikan kontainer YARN untuk itu. Jika simpul tetap tidak sehat selama 45 menit, Amazon EMR menandai instans Amazon EC2 terkait untuk diakhiri sebagai `FAILED_BY_MASTER`. Ketika semua instans Amazon EC2 yang terkait dengan simpul inti ditandai untuk pengakhiran, klaster berakhir dengan status `NO_SLAVE_LEFT` karena tidak ada sumber daya untuk melaksanakan pekerjaan.

Melebihi pemanfaatan disk pada satu simpul inti mungkin menyebabkan reaksi berantai. Jika satu simpul melebihi ambang pemanfaatan disk akibat HDFS, simpul lain kemungkinan besar juga berada dekat ambang. Node pertama melebihi ambang batas pemanfaatan disk, jadi Amazon EMR menolak mencantumkannya. Hal ini meningkatkan beban pemanfaatan disk untuk node yang tersisa karena mereka mulai mereplikasi data HDFS di antara mereka sendiri yang hilang pada node deny-listed. Setiap simpul kemudian menjadi `UNHEALTHY` dengan cara yang sama, dan klaster akhirnya berakhir.

Praktik terbaik dan rekomendasi

Mengkonfigurasi perangkat keras klaster dengan penyimpanan yang memadai

Ketika Anda membuat sebuah klaster, pastikan bahwa ada cukup simpul inti dan masing-masing memiliki penyimpanan instans serta volume penyimpanan EBS yang memadai untuk HDFS. Untuk informasi selengkapnya, lihat Menghitung kapasitas HDFS yang dibutuhkan dari sebuah klaster. Anda juga dapat menambahkan instans inti ke grup instans yang ada secara manual atau dengan menggunakan penskalaan otomatis. Instans yang baru memiliki konfigurasi penyimpanan yang sama seperti instans lain dalam grup instans. Untuk informasi selengkapnya, lihat Gunakan penskalaan cluster.

Aktifkan perlindungan penghentian

Aktifkan perlindungan penghentian Dengan cara ini, jika node inti ditolak terdaftar, Anda dapat terhubung ke instans Amazon EC2 terkait menggunakan SSH untuk memecahkan masalah dan memulihkan data. Jika Anda mengaktifkan proteksi pengakhiran, harap diingat bahwa Amazon EMR tidak menggantikan instans Amazon EC2 dengan instans baru. Untuk informasi selengkapnya, lihat Menggunakan perlindungan pengakhiran.

Buat alarm untuk UnhealthyNodes CloudWatch metrik MR

Metrik ini melaporkan jumlah simpul yang melaporkan Status UNHEALTHY. Ini setara dengan metrik YARN `mapred.resourcemanager.NoOfUnhealthyNodes`. Anda dapat mengatur notifikasi untuk alarm ini agar memperingatkan Anda tentang simpul yang tidak sehat sebelum batas waktu 45 menit tercapai. Untuk informasi selengkapnya, lihat Memantau metrik Amazon EMR dengan CloudWatch.

Ubah pengaturan menggunakan situs yarn

Pengaturan di bawah ini dapat disesuaikan sesuai dengan kebutuhan aplikasi Anda. Sebagai contoh, Anda mungkin ingin meningkatkan ambang pemanfaatan disk tempat simpul melaporkan UNHEALTHY dengan meningkatkan nilai `yarn.nodemanager.disk-health-checker.max-disk-utilization-per-disk-percentage`.

Anda dapat mengatur nilai-nilai ini ketika Anda membuat sebuah klaster menggunakan klasifikasi konfigurasi `yarn-site`. Untuk informasi selengkapnya, lihat Mengkonfigurasi Aplikasi dalam Panduan Rilis Amazon EMR. Anda juga dapat menyambung ke instans Amazon EC2 yang terkait dengan simpul inti menggunakan SSH, dan kemudian menambahkan nilai-nilai dalam `/etc/hadoop/conf.empty/yarn-site.xml` menggunakan editor teks. Setelah melakukan perubahan, Anda harus memulai ulang hadoop-yarn-nodemanager seperti yang ditunjukkan di bawah ini.

> **Important**
>
> Saat Anda memulai ulang NodeManager layanan, kontainer YARN aktif dimatikan kecuali `yarn.nodemanager.recovery.enabled` diatur untuk `true` menggunakan klasifikasi `yarn-site` konfigurasi saat Anda membuat cluster. Anda juga harus menentukan direktori tempat menyimpan status kontainer menggunakan properti `yarn.nodemanager.recovery.dir`.

```
sudo /sbin/stop hadoop-yarn-nodemanager
sudo /sbin/start hadoop-yarn-nodemanager
```

Untuk informasi selengkapnya tentang properti dan nilai default `yarn-site` saat ini, lihat Pengaturan default YARN dalam dokumentasi Apache Hadoop.

Properti	Nilai default	Deskripsi
benang.nodemanager. disk-health-checker.interval-ms	120000	Frekuensi (dalam detik) yang dijalankan oleh pemeriksa kesehatan disk.
benang.nodemanager. disk-health-checker. min-healthy-disks	0,25	Fraksi minimum dari jumlah disk yang harus sehat NodeManager untuk meluncurkan wadah baru. Hal ini sesuai dengan kedua yarn.nodemanager.local-dirs (secara default, `/mnt/yarn` di Amazon EMR) dan yarn.nodemanager.log-dirs (secara default `/var/log/hadoop-yarn/containers`, yang symlinked ke `mnt/var/log/hadoop-yarn/containers` di Amazon EMR).
`yarn.nodemanager.disk-health-checker.max-disk-utilization-per-disk-percentage`	90.0	Persentase maksimum pemanfaatan ruang disk yang diizinkan setelah disk ditandai sebagai buruk. Nilai dapat berkisar dari 0,0 hingga 100,0. Jika nilainya lebih besar dari atau sama dengan 100, NodeManager cek untuk disk penuh. Ini berlaku baik untuk `yarn-nodemanager.local-dirs` dan `yarn.nodemanager.log-dirs`.

Properti	Nilai default	Deskripsi
`yarn.nodemanager.disk-health-checker.min-free-space-per-disk-mb`	0	Ruang minimum yang mesti tersedia pada disk agar dapat digunakan. Ini berlaku baik untuk `yarn-nodemanager.local-dirs` dan `yarn.nodemanager.log-dirs` .

Tidak dapat mereplikasi blok, hanya berhasil mereplikasi ke nol simpul.

Kesalahan, “Tidak dapat mereplikasi blok, hanya berhasil mereplikasi ke nol simpul.” biasanya terjadi ketika sebuah klaster tidak memiliki penyimpanan HDFS yang cukup. Kesalahan ini terjadi ketika Anda menghasilkan lebih banyak data di klaster Anda daripada yang dapat disimpan dalam HDFS. Anda melihat kesalahan ini hanya saat klaster berjalan, karena ketika pekerjaan berakhir klaster akan merilis ruang HDFS yang digunakan.

Jumlah ruang HDFS yang tersedia untuk klaster bergantung pada jumlah dan tipe instans Amazon EC2 yang digunakan sebagai simpul inti. Simpul tugas tidak digunakan untuk penyimpanan HDFS. Semua ruang disk pada setiap instans Amazon EC2, termasuk volume penyimpanan EBS terpasang, tersedia untuk HDFS. Untuk informasi selengkapnya tentang jumlah penyimpanan lokal untuk setiap tipe instans EC2, lihat Jenis dan keluarga instans di Panduan Pengguna Amazon EC2 untuk Instans Linux.

Faktor lain yang dapat mempengaruhi jumlah ruang HDFS yang tersedia adalah faktor replikasi, yang merupakan jumlah salinan dari setiap blok data yang disimpan dalam HDFS untuk redundansi. Faktor replikasi meningkat dengan jumlah simpul dalam klaster: ada 3 salinan dari setiap blok data untuk klaster dengan 10 simpul atau lebih, 2 salinan dari setiap blok untuk klaster dengan 4 sampai 9 simpul, dan 1 salinan (tidak ada redundansi) untuk klaster dengan 3 simpul atau kurang. Total ruang HDFS yang tersedia dibagi dengan faktor replikasi. Dalam beberapa kasus, seperti meningkatkan jumlah simpul dari 9 ke 10, peningkatan faktor replikasi dapat benar-benar menyebabkan jumlah ruang HDFS yang tersedia berkurang.

Sebagai contoh, sebuah klaster dengan sepuluh simpul inti tipe m1.large akan memiliki 2833 GB ruang yang tersedia untuk HDFS ((10 simpul X 850 GB per simpul)/faktor replikasi 3).

Jika klaster melebihi jumlah ruang yang tersedia untuk HDFS, Anda dapat menambahkan simpul inti tambahan untuk klaster Anda atau menggunakan kompresi data untuk membuat lebih banyak ruang HDFS. Jika klaster Anda dapat dihentikan dan dimulai ulang, Anda dapat mempertimbangkan menggunakan simpul inti tipe instans Amazon EC2 yang lebih besar. Anda juga dapat mempertimbangkan menyesuaikan faktor replikasi. Namun, harap diingat bahwa penurunan faktor replikasi akan mengurangi redundansi data HDFS dan kemampuan pemulihan klaster Anda dari kehilangan atau kerusakan blok HDFS.

KUOTA EC2 TERLAMPAUI

Jika Anda mendapatkan pesan `EC2 QUOTA EXCEEDED`, mungkin ada beberapa penyebab. Tergantung pada perbedaan konfigurasi, mungkin diperlukan waktu hingga 5-20 menit bagi klaster sebelumnya untuk berakhir dan melepaskan sumber daya yang dialokasikan. Jika Anda mendapatkan kesalahan `EC2 QUOTA EXCEEDED` ketika Anda mencoba untuk meluncurkan sebuah klaster, mungkin itu karena sumber daya dari klaster yang baru berakhir belum dirilis. Pesan ini juga dapat disebabkan oleh perubahan ukuran grup instans atau armada instans ke ukuran target yang lebih besar daripada kuota instans saat ini untuk akun tersebut. Hal ini dapat terjadi secara manual atau otomatis melalui penskalaan otomatis.

Pertimbangkan opsi berikut untuk menyelesaikan masalah:

- Ikuti petunjuk dalam kuota AWS layanan di Referensi Umum Amazon Webuntuk meminta peningkatan batas layanan. Untuk beberapa API, menyiapkan CloudWatch acara mungkin merupakan pilihan yang lebih baik daripada meningkatkan batas. Untuk detail selengkapnya, lihat Waktu untuk mengatur kejadian EMR di CloudWatch.
- Jika satu klaster atau lebih yang sedang berjalan tidak pada kapasitas yang ditentukan, ubah ukuran grup instans atau kurangi kapasitas target pada armada instans untuk klaster yang sedang berjalan.
- Membuat klaster dengan lebih sedikit instans EC2 atau kapasitas target yang lebih rendah.

Terlalu banyak kegagalan mengambil

Adanya pesan kesalahan ”Terlalu banyak kegagalan mengambil“ atau ”Kesalahan saat membaca output tugas“ dalam log langkah atau upaya tugas menunjukkan bahwa tugas tersebut bergantung pada output tugas lain. Hal ini sering terjadi ketika tugas peredaman berada dalam antrean untuk dieksekusi dan membutuhkan output dari satu atau lebih tugas pemetaan dan outputnya belum tersedia.

Ada beberapa alasan output mungkin tidak tersedia:

- Tugas prasyarat masih memproses. Hal ini seringkali berupa tugas pemetaan.
- Data mungkin tidak tersedia karena konektivitas jaringan yang buruk jika data terletak pada instans yang berbeda.
- Jika HDFS digunakan untuk mengambil output, mungkin ada masalah dengan HDFS.

Penyebab paling umum dari kesalahan ini adalah bahwa tugas sebelumnya masih diproses. Hal ini mungkin terjadi jika kesalahan muncul saat tugas peredaman mencoba berjalan untuk pertama kalinya. Anda dapat memeriksa apakah hal ini yang terjadi dengan meninjau log syslog untuk langkah klaster yang mengembalikan kesalahan. Jika syslog menunjukkan adanya kemajuan tugas pemetaan dan peredaman, ini menunjukkan bahwa fase peredaman telah dimulai saat ada tugas pemetaan yang belum selesai.

Satu hal yang harus dicari dalam log adalah persentase kemajuan pemetaan yang mencapai 100% dan kemudian turun kembali ke nilai yang lebih rendah. Ketika persentase pemetaan mencapai 100%, ini tidak berarti bahwa semua tugas pemetaan selesai. Ini hanya berarti bahwa Hadoop mengeksekusi semua tugas pemetaan. Jika nilai ini turun kembali di bawah 100%, itu berarti bahwa tugas pemetaan telah gagal dan, tergantung pada konfigurasi, Hadoop dapat mencoba untuk menjadwalkan ulang tugas tersebut. Jika persentase peta tetap 100% di log, lihat CloudWatch metrik, khususnya`RunningMapTasks`, untuk memeriksa apakah tugas peta masih diproses. Anda juga dapat menemukan informasi ini menggunakan antarmuka web Hadoop pada simpul utama.

Jika Anda melihat masalah ini, ada beberapa hal yang dapat Anda coba:

- Instruksikan fase peredaman untuk menunggu lebih lama sebelum mulai. Anda dapat melakukan ini dengan mengubah pengaturan konfigurasi Hadoop mapred.reduce.slowstart.completed.maps ke waktu yang lebih lama. Untuk informasi selengkapnya, lihat Buat tindakan bootstrap untuk menginstal perangkat lunak tambahan.
- Cocokkan jumlah peredam dengan kemampuan peredam total di klaster tersebut. Anda melakukan ini dengan menyesuaikan pengaturan konfigurasi Hadoop mapred.reduce.tasks untuk pekerjaan tersebut.
- Gunakan kode kelas combiner untuk meminimalisir jumlah output yang perlu diambil.
- Periksa bahwa tidak ada masalah dengan layanan Amazon EC2 yang mempengaruhi performa jaringan klaster. Anda dapat melakukannya menggunakan Service Health Dashboard.

- Meninjau sumber daya CPU dan memori instans di klaster Anda untuk memastikan bahwa pemrosesan data Anda tidak membuat sumber daya simpul Anda kewalahan. Untuk informasi selengkapnya, lihat Konfigurasi perangkat keras dan jaringan klaster.
- Periksa versi Amazon Machine Image (AMI) yang digunakan dalam klaster Amazon EMR Anda. Jika versinya adalah versi 2.3.0 hingga 2.4.4 inklusif, perbarui ke versi yang lebih baru. Versi AMI dalam kisaran tertentu menggunakan versi Jetty yang mungkin gagal memberikan output dari fase pemetaan. Kesalahan pengambilan terjadi ketika peredam tidak dapat memperoleh output dari fase pemetaan.

  Jetty adalah server HTTP sumber terbuka yang digunakan untuk komunikasi antar mesin dalam klaster Hadoop.

File hanya dapat direplikasi ke 0 simpul bukan 1

Ketika file ditulis ke HDFS, file tersebut direplikasi ke beberapa simpul inti. Ketika Anda melihat kesalahan ini, itu berarti bahwa NameNode daemon tidak memiliki DataNode instance yang tersedia untuk menulis data dalam HDFS. Dengan kata lain, tidak terjadi replikasi blok. Kesalahan ini dapat disebabkan oleh sejumlah masalah:

- Sistem berkas HDFS mungkin telah kehabisan ruang. Ini adalah penyebab yang paling mungkin.
- DataNode Contoh mungkin tidak tersedia saat pekerjaan dijalankan.
- DataNode instance mungkin telah diblokir dari komunikasi dengan node master.
- Instans dalam grup instans inti mungkin tidak tersedia.
- Izin mungkin hilang. Misalnya, JobTracker daemon mungkin tidak memiliki izin untuk membuat informasi pelacak pekerjaan.
- Pengaturan ruang cadangan untuk sebuah DataNode instans mungkin tidak cukup. Periksa apakah hal ini yang terjadi dengan memeriksa pengaturan konfigurasi dfs.datanode.du.reserved.

Untuk memeriksa apakah masalah ini disebabkan oleh HDFS kehabisan ruang disk, lihat `HDFSUtilization` metrik di CloudWatch. Jika nilai ini terlalu tinggi, Anda dapat menambahkan simpul inti tambahan untuk klaster tersebut. Jika Anda memiliki cluster yang menurut Anda mungkin kehabisan ruang disk HDFS, Anda dapat mengatur alarm CloudWatch untuk mengingatkan Anda ketika nilai `HDFSUtilization` naik di atas tingkat tertentu. Untuk informasi selengkapnya, silakan lihat Secara manual mengubah ukuran klaster berjalan dan Memantau metrik Amazon EMR dengan CloudWatch.

Jika HDFS kehabisan ruang bukan masalahnya, periksa log, DataNode log, dan konektivitas jaringan untuk masalah lain yang dapat mencegah HDFS mereplikasi data. NameNode Untuk informasi selengkapnya, lihat Melihat berkas log.

Node yang terdaftar penolakan

NodeManager Daemon bertanggung jawab untuk meluncurkan dan mengelola kontainer pada node inti dan tugas. Kontainer dialokasikan ke NodeManager daemon oleh ResourceManager daemon yang berjalan pada node master. ResourceManager Memonitor NodeManager simpul melalui detak jantung.

Ada beberapa situasi di mana ResourceManager daemon menolak mencantumkan a NodeManager, menghapusnya dari kumpulan node yang tersedia untuk memproses tugas:

- Jika NodeManager belum mengirim detak jantung ke ResourceManager daemon dalam 10 menit terakhir (600.000 milidetik). Periode waktu ini dapat dikonfigurasi menggunakan pengaturan konfigurasi `yarn.nm.liveness-monitor.expiry-interval-ms`. Untuk informasi selengkapnya tentang mengubah klasifikasi konfigurasi Yarn, lihat Mengkonfigurasi aplikasi dalam Panduan Rilis Amazon EMR.
- NodeManager memeriksa kesehatan disk yang ditentukan oleh `yarn.nodemanager.local-dirs` dan`yarn.nodemanager.log-dirs`. Pemeriksaan ini termasuk izin dan ruang disk kosong (< 90%). Jika disk gagal memeriksa, NodeManager berhenti menggunakan disk tertentu tetapi masih melaporkan status node sebagai sehat. Jika sejumlah disk gagal dalam pemeriksaan, node dilaporkan tidak sehat ke wadah baru ResourceManager dan wadah baru tidak ditetapkan ke node.

Master aplikasi juga dapat menolak daftar NodeManager node jika memiliki lebih dari tiga tugas yang gagal. Anda dapat mengubah ini ke nilai yang lebih tinggi menggunakan parameter konfigurasi `mapreduce.job.maxtaskfailures.per.tracker`. Pengaturan konfigurasi lain yang mungkin Anda ubah akan mengendalikan berapa kali percobaan tugas dilakukan sebelum menandainya sebagai gagal: `mapreduce.map.max.attempts` untuk tugas pemetaan dan `mapreduce.reduce.maxattempts` untuk tugas peredaman. Untuk informasi selengkapnya tentang mengubah klasifikasi konfigurasi, lihat Mengkonfigurasi aplikasi dalam Panduan Rilis Amazon EMR.

Kesalahan throttling

Kesalahan “Terjadi throttling dari *Amazon EC2* saat meluncurkan klaster” dan “Gagal menyediakan instans akibat throttling dari *Amazon EC2*“ terjadi saat Amazon EMR tidak dapat menyelesaikan

permintaan karena layanan lain telah men-throttle aktivitas. Amazon EC2 adalah sumber yang paling umum dari kesalahan throttling, tetapi layanan lain mungkin menjadi penyebab kesalahan throttling. AWS service limits berlaku berdasarkan wilayah untuk meningkatkan performa, dan kesalahan throttling menunjukkan bahwa Anda telah melampaui batas layanan untuk akun Anda di Wilayah tersebut.

Kemungkinan penyebab

Sumber yang paling umum dari kesalahan throttling Amazon EC2 adalah sejumlah besar instans klaster yang diluncurkan sehingga batas layanan Anda untuk instans EC2 terlampaui. Instans klaster dapat diluncurkan karena alasan berikut:

- Klaster baru dibuat.
- Klaster diubah ukurannya secara manual. Untuk informasi selengkapnya, lihat Secara manual mengubah ukuran klaster berjalan.
- Grup instans dalam sebuah klaster menambahkan instans (menskalakan keluar) sebagai hasil dari aturan penskalaan otomatis. Untuk informasi selengkapnya, lihat Memahami aturan penskalaan otomatis.
- Armada instans dalam sebuah klaster menambahkan instans untuk memenuhi kapasitas target yang meningkat. Untuk informasi selengkapnya, lihat Mengkonfigurasi armada instans.

Kemungkinan lain adalah bahwa frekuensi atau jenis permintaan API yang dibuat untuk Amazon EC2 mengakibatkan kesalahan throttling. Untuk informasi lebih lanjut tentang cara Amazon EC2 melakukan throttling permintaan API, lihat Tingkat permintaan API kueri di Referensi API Amazon EC2.

Solusi

Pertimbangkan solusi berikut:

- Ikuti petunjuk dalam kuota AWS layanan di Referensi Umum Amazon Webuntuk meminta peningkatan batas layanan. Untuk beberapa API, menyiapkan CloudWatch acara mungkin merupakan pilihan yang lebih baik daripada meningkatkan batas. Untuk detail selengkapnya, lihat Waktu untuk mengatur kejadian EMR di CloudWatch.
- Jika Anda memiliki klaster yang diluncurkan pada jadwal yang sama—misalnya, di awal jam—pertimbangkan waktu mulai yang mengejutkan.
- Jika Anda memiliki klaster yang diberi ukuran untuk permintaan puncak, dan Anda secara berkala memiliki kapasitas instans, pertimbangkan untuk menentukan penskalaan otomatis untuk

menambahkan dan menghapus instans sesuai permintaan. Dengan cara ini, instans digunakan secara lebih efisien, dan tergantung pada profil permintaan, kemungkinan lebih sedikit instans yang diminta pada waktu tertentu di akun. Untuk informasi selengkapnya, lihat Menggunakan penskalaan otomatis dengan kebijakan kustom untuk grup instans .

Tipe instans tidak didukung

Jika Anda membuat klaster, dan gagal dengan pesan kesalahan "Jenis instans yang diminta tidak *InstanceType*didukung di Zona Ketersediaan yang diminta," artinya Anda membuat klaster dan menetapkan jenis instans untuk satu atau beberapa grup instans yang tidak didukung oleh EMR Amazon di Wilayah dan Zona Ketersediaan tempat klaster dibuat. Amazon EMR dapat mendukung sebuah tipe instans di satu Availability Zone dalam suatu Wilayah dan tidak di wilayah lain. Subnet yang Anda pilih untuk sebuah klaster menentukan Availability Zone di dalam Wilayah tersebut.

Solusi

Menentukan tipe instans yang tersedia di Availability Zone menggunakan AWS CLI

- Gunakan perintah `ec2 run-instances` dengan opsi `--dry-run`. Pada contoh di bawah ini, ganti *m5.xlarge* dengan tipe instans yang ingin Anda gunakan, *ami-035be7bafff33b6b6* dengan AMI yang terkait dengan tipe instans tersebut, dan *subnet-12ab3c45* dengan subnet di Availability Zone yang ingin Anda kueri.

  ```
  aws ec2 run-instances --instance-type m5.xlarge --dry-run --image-
  id ami-035be7bafff33b6b6 --subnet-id subnet-12ab3c45
  ```

  Untuk petunjuk tentang menemukan ID AMI, lihat Temukan AMI Linux. Untuk menemukan ID subnet, Anda dapat menggunakan perintah describe-subnets.

Untuk mempelajari lebih lanjut tentang cara menemukan jenis instans yang tersedia, lihat Menemukan jenis instans Amazon EC2.

Setelah Anda menentukan tipe instans yang tersedia, Anda dapat melakukan hal-hal berikut:

- Buat klaster di Wilayah dan Subnet EC2 yang sama, dan pilih tipe instans yang berbeda dengan kemampuan yang sama dengan pilihan awal Anda. Untuk daftar tipe instans yang didukung, lihat Tipe instans yang didukung. Untuk membandingkan kemampuan tipe instans EC2, lihat Tipe instans Amazon EC2.

- Pilih subnet untuk klaster di Availability Zone tempat tipe instans tersebut tersedia dan didukung oleh Amazon EMR.

EC2 berada di luar kapasitas

Kesalahan “EC2 berada di luar kapasitas untuk *InstanceType*” terjadi ketika Anda mencoba membuat klaster, atau menambahkan instance ke cluster, di Availability Zone yang tidak memiliki jenis instans EC2 yang ditentukan lagi. Subnet yang Anda pilih untuk sebuah klaster menentukan Availability Zone.

Untuk membuat klaster baru, lakukan salah satu dari hal berikut:

- Tentukan tipe instans yang berbeda dengan kemampuan serupa
- Buat klaster di Wilayah yang berbeda
- Pilih subnet di Availability Zone tempat tipe instans yang Anda inginkan mungkin tersedia.

Untuk menambahkan instans ke klaster yang sedang berjalan, lakukan salah satu hal berikut:

- Ubah konfigurasi grup instans atau konfigurasi armada instans untuk menambahkan tipe instans yang tersedia dengan kemampuan serupa. Untuk daftar tipe instans yang didukung, lihat Tipe instans yang didukung. Untuk membandingkan kemampuan tipe instans EC2, lihat Tipe instans Amazon EC2.
- Akhiri klaster dan buat kembali klaster tersebut di Availability Zone tempat tipe instans tersebut tersedia.

Kesalahan input dan output

Kesalahan berikut umum terjadi dalam operasi input dan output klaster.

Topik

- Apakah jalur Anda ke Amazon Simple Storage Service (Amazon S3) memiliki setidaknya tiga garis miring?
- Apakah Anda mencoba untuk melintasi direktori input secara rekursif?
- Apakah direktori output Anda sudah ada?
- Apakah Anda mencoba menentukan sumber daya menggunakan URL HTTP?
- Apakah Anda mereferensikan bucket Amazon S3 menggunakan format nama yang tidak valid?

- Apakah Anda mengalami kesulitan memuat data ke atau dari Amazon S3?

Apakah jalur Anda ke Amazon Simple Storage Service (Amazon S3) memiliki setidaknya tiga garis miring?

Ketika Anda menentukan bucket Amazon S3, Anda harus menyertakan garis miring yang mengakhiri pada akhir URL. Misalnya, alih-alih mereferensikan bucket sebagai “s3n://*DOC-EXAMPLE-BUCKET1*“, Anda harus menggunakan “s3n://*DOC-EXAMPLE-BUCKET1*/”, jika tidak Hadoop biasanya akan menggagalkan klaster Anda.

Apakah Anda mencoba untuk melintasi direktori input secara rekursif?

Hadoop tidak mencari direktori input untuk file secara rekursif. Jika Anda memiliki struktur direktori seperti /corpus/01/01.txt, /corpus/01/02.txt, /corpus/02/01.txt, dll dan Anda menentukan /corpus/ sebagai parameter input ke klaster Anda, Hadoop tidak menemukan file input karena direktori / corpus/ kosong dan Hadoop tidak memeriksa isi dari subdirektori. Demikian pula, Hadoop tidak memeriksa subdirektori bucket Amazon S3 secara rekursif.

File input harus langsung dalam direktori input atau bucket Amazon S3 yang Anda tentukan, bukan di dalam subdirektori.

Apakah direktori output Anda sudah ada?

Jika Anda menentukan jalur output yang sudah ada, Hadoop biasanya akan menggagalkan klaster. Ini berarti bahwa jika Anda menjalankan sebuah klaster satu kali dan kemudian menjalankannya lagi dengan parameter yang persis sama, klaster tersebut mungkin akan berjalan pada awalnya namun tidak akan berjalan kembali; setelah dijalankan pertama kali, tersedia jalur output dan karenanya menyebabkan yang dijalankan setelah itu gagal.

Apakah Anda mencoba menentukan sumber daya menggunakan URL HTTP?

Hadoop tidak menerima lokasi sumber daya yang ditentukan menggunakan prefiks http://. Anda tidak dapat mereferensikan sumber daya menggunakan URL HTTP. Sebagai contoh, menggunakan http://mysite/myjar.jar sebagai parameter JAR akan menyebabkan klaster gagal.

Apakah Anda mereferensikan bucket Amazon S3 menggunakan format nama yang tidak valid?

Jika Anda mencoba menggunakan nama bucket seperti ”*DOC-EXAMPLE-BUCKET1*.1" dengan Amazon EMR, klaster Anda akan gagal karena Amazon EMR mengharuskan nama bucket valid

berupa nama host RFC 2396; nama tidak dapat berakhiran nomor. Selain itu, karena persyaratan Hadoop, nama bucket Amazon S3 yang digunakan dengan Amazon EMR harus berisi huruf kecil, angka, titik (.), dan tanda hubung (-) saja. Untuk informasi selengkapnya tentang cara memformat nama bucket Amazon S3, lihat Pembatasan dan batasan bucket di Panduan Pengguna Layanan Penyimpanan Sederhana Amazon.

Apakah Anda mengalami kesulitan memuat data ke atau dari Amazon S3?

Amazon S3 adalah sumber input dan output paling populer untuk Amazon EMR. Kesalahan umum yang terjadi adalah memperlakukan Amazon S3 seperti yang Anda memperlakukan sistem file yang umum. Ada perbedaan antara Amazon S3 dan sistem file yang perlu Anda pertimbangkan saat menjalankan klaster Anda.

- Jika terjadi kesalahan internal di Amazon S3, aplikasi Anda perlu menangani ini secara perlahan dan mencoba kembali operasi.
- Jika panggilan ke Amazon S3 memakan waktu terlalu lama untuk dikembalikan, aplikasi Anda mungkin perlu mengurangi frekuensi pemanggilan Amazon S3.
- Mencantumkan semua objek dalam bucket Amazon S3 adalah panggilan yang mahal. Aplikasi Anda harus meminimalisir jumlah hal ini dilakukan.

Ada beberapa cara Anda dapat meningkatkan cara klaster Anda berinteraksi dengan Amazon S3.

- Luncurkan klaster Anda menggunakan versi rilis terbaru dari Amazon EMR.
- Gunakan S3 DistCp untuk memindahkan objek masuk dan keluar dari Amazon S3. S3 DistCp mengimplementasikan penanganan kesalahan, percobaan ulang, dan back-off agar sesuai dengan persyaratan Amazon S3. Untuk informasi selengkapnya, lihat Salinan terdistribusi menggunakan S3 DistCp.
- Desain aplikasi Anda dengan mempertimbangkan eventual consistency. Gunakan HDFS untuk penyimpanan data menengah saat klaster berjalan dan Amazon S3 hanya untuk memasukkan data awal dan mengeluarkan hasil akhir.
- Jika klaster Anda akan melakukan 200 atau lebih transaksi per detik ke Amazon S3, kontak support untuk mempersiapkan bucket Anda untuk transaksi per detik yang lebih besar dan pertimbangkan untuk menggunakan strategi partisi kunci yang dijelaskan di Tips & trik performa Amazon S3.
- Mengatur konfigurasi Hadoop pengaturan io.file.buffer.size menjadi 65536. Hal ini menyebabkan Hadoop untuk menghabiskan lebih sedikit waktu mencari melalui objek Amazon S3.

- Pertimbangkan menonaktifkan fitur eksekusi spekulatif Hadoop jika klaster Anda mengalami masalah konkurensi Amazon S3. Hal ini juga berguna ketika Anda memecahkan masalah klaster lambat. Anda melakukan ini dengan menetapkan properti `mapreduce.map.speculative` dan `mapreduce.reduce.speculative` menjadi `false`. Ketika Anda meluncurkan klaster, Anda dapat mengatur nilai ini menggunakan klasifikasi konfigurasi `mapred-env`. Untuk informasi selengkapnya, lihat Mengkonfigurasi Aplikasi dalam Panduan Rilis Amazon EMR.
- Jika Anda menjalankan klaster Hive, lihat Apakah Anda mengalami kesulitan memuat data ke atau dari Amazon S3 ke Hive?.

Untuk informasi tambahan, lihat praktik terbaik kesalahan Amazon S3 di Panduan Pengguna Layanan Penyimpanan Sederhana Amazon.

Kesalahan izin

Kesalahan berikut umum terjadi ketika menggunakan izin atau kredensial.

Topik

- Apakah Anda meloloskan kredensial yang benar ke SSH?
- Jika Anda menggunakan IAM, apakah Anda telah menetapkan kebijakan Amazon EC2 yang benar?

Apakah Anda meloloskan kredensial yang benar ke SSH?

Jika Anda tidak dapat menggunakan SSH untuk tersambung ke simpul utama, kemungkinan besar ada masalah dengan kredensial keamanan Anda.

Pertama, periksa bahwa file .pem yang berisi kunci SSH Anda memiliki izin yang tepat. Anda dapat menggunakan chmod untuk mengubah izin pada file .pem Anda seperti yang ditunjukkan dalam contoh berikut, dimana Anda akan mengganti mykey.pem dengan nama file .pem Anda sendiri.

```
chmod og-rwx mykey.pem
```

Kemungkinan kedua adalah bahwa Anda tidak menggunakan pasangan kunci yang Anda tentukan saat Anda membuat klaster. Hal ini mudah dilakukan jika Anda telah membuat beberapa pasangan

kunci. Periksa detail klaster di konsol Amazon EMR (atau gunakan opsi `--describe` di CLI) untuk nama pasangan kunci yang ditentukan ketika klaster dibuat.

Setelah Anda telah memverifikasi bahwa Anda menggunakan pasangan kunci yang benar dan izin telah ditetapkan dengan benar pada file .pem, Anda dapat menggunakan perintah berikut untuk menggunakan SSH untuk menyambung ke simpul utama, dimana Anda akan mengganti mykey.pem dengan nama file .pem dan `hadoop@ec2-01-001-001-1.compute-1.amazonaws.com` dengan nama DNS publik simpul utama (tersedia melalui opsi `--describe` di CLI atau melalui konsol Amazon EMR.)

> **Important**
>
> Anda harus menggunakan nama login `hadoop` ketika Anda menyambungkan ke simpul klaster Amazon EMR, jika tidak maka kesalahan yang mirip dengan `Server refused our key` mungkin terjadi.

```
ssh -i mykey.pem hadoop@ec2-01-001-001-1.compute-1.amazonaws.com
```

Untuk informasi selengkapnya, lihat Connect ke node utama menggunakan SSH.

Jika Anda menggunakan IAM, apakah Anda telah menetapkan kebijakan Amazon EC2 yang benar?

Karena Amazon EMR menggunakan instans EC2 sebagai node, pengguna Amazon EMR juga perlu memiliki kebijakan Amazon EC2 tertentu yang ditetapkan agar Amazon EMR dapat mengelola instans tersebut atas nama pengguna. Jika Anda tidak memiliki izin yang diperlukan, Amazon EMR mengembalikan kesalahan: “akun tidak diizinkan untuk memanggil EC2.”

Untuk informasi selengkapnya tentang kebijakan Amazon EC2 yang harus ditetapkan IAM Anda untuk menjalankan Amazon EMR, lihat Cara kerja Amazon EMR dengan IAM.

Kesalahan Klaster Hive

Anda biasanya dapat menemukan penyebab kesalahan Hive di file `syslog`, yang Anda tautkan dari panel Langkah. Jika Anda tidak dapat menentukan masalahnya di sana, periksa pesan kesalahan Hadoop upaya tugas. Tautkan ke sana pada panel Upaya Tugas.

Kesalahan berikut umum terjadi untuk klaster Hive.

Topik

Apakah Anda menggunakan versi terbaru dari Hive?

Versi terbaru dari Hive memiliki semua patch dan perbaikan bug terbaru dan dapat menyelesaikan masalah Anda.

Apakah Anda mengalami kesalahan sintaks dalam skrip Hive?

Jika langkah gagal, lihat file `stdout` log untuk langkah yang menjalankan skrip Hive. Jika kesalahan tidak ada di sana, lihat file `syslog` log upaya tugas untuk upaya tugas yang gagal. Untuk informasi selengkapnya, lihat Melihat berkas log.

Apakah pekerjaan gagal saat berjalan secara interaktif?

Jika Anda menjalankan Hive secara interaktif pada simpul utama dan klaster tersebut gagal, lihat entri `syslog` dalam log upaya tugas untuk upaya tugas yang gagal. Untuk informasi selengkapnya, lihat Melihat berkas log.

Apakah Anda mengalami kesulitan memuat data ke atau dari Amazon S3 ke Hive?

Jika Anda mengalami kesulitan mengakses data di Amazon S3, periksa dulu kemungkinan penyebab yang tercantum dalam Apakah Anda mengalami kesulitan memuat data ke atau dari Amazon S3?. Jika masalah yang ada di sana bukan penyebabnya, pertimbangkan opsi berikut khusus untuk Hive.

- Pastikan Anda menggunakan versi Hive terbaru, yang memiliki semua patch dan perbaikan bug terbaru yang dapat menyelesaikan masalah Anda. Untuk informasi selengkapnya, lihat Apache Hive.
- Menggunakan `INSERT OVERWRITE` memerlukan pencantuman isi bucket atau folder Amazon S3. Ini adalah operasi yang mahal. Jika memungkinkan, pangkas jalurnya secara manual alih-alih membuat Hive mencantumkan dan menghapus objek yang ada.

- Jika Anda menggunakan versi riliis Amazon EMR yang lebih tua dari 5.0, Anda dapat menggunakan perintah berikut di HiveQL untuk melakukan pra-cache hasil operasi pencantuman Amazon S3 secara lokal di klaster:

```
set hive.optimize.s3.query=true;
```

- Gunakan partisi statis jika memungkinkan.
- Dalam beberapa versi Hive dan Amazon EMR, penggunaan ALTER TABEL mungkin akan gagal karena tabel disimpan di lokasi yang berbeda dari yang diharapkan oleh Hive. Solusinya adalah menambahkan atau memperbarui hal berikut di `/home/hadoop/conf/core-site.xml`:

```
<property>
    <name>fs.s3n.endpoint</name>
    <value>s3.amazonaws.com</value>
</property>
```

Kesalahan VPC

Kesalahan berikut umum terjadi untuk konfigurasi VPC di Amazon EMR.

Topik

- Konfigurasi subnet tidak valid
- Set Opsi DHCP Hilang
- Kesalahan izin
- Kesalahan yang mengakibatkan START_FAILED
- Cluster Terminated with errors dan NameNode gagal memulai

Konfigurasi subnet tidak valid

Pada halaman Detail Klaster, di bidang Status, Anda melihat kesalahan yang mirip dengan yang berikut ini:

`The subnet configuration was invalid: Cannot find route to InternetGateway in main RouteTable` *`rtb-id`* `for vpc` *`vpc-id`*`.`

Untuk mengatasi masalah ini, Anda harus membuat Gateway Internet dan melampirkannya ke VPC Anda. Untuk informasi selengkapnya, lihat Menambahkan gateway internet ke VPC Anda.

Atau, verifikasi bahwa Anda telah mengkonfigurasi VPC Anda dengan Aktifkan resolusi DNS dan Aktifkan dukungan nama host DNS diaktifkan. Untuk informasi selengkapnya, lihat Menggunakan DNS dengan VPC Anda.

Set Opsi DHCP Hilang

Anda melihat kegagalan langkah dalam log sistem klaster (syslog) dengan kesalahan yang mirip dengan yang berikut ini:

```
ERROR org.apache.hadoop.security.UserGroupInformation
(main): PriviledgedActionException as:hadoop (auth:SIMPLE)
cause:java.io.IOException:
org.apache.hadoop.yarn.exceptions.ApplicationNotFoundException: Application
with id 'application_id' doesn't exist in RM.
```

atau

```
ERROR org.apache.hadoop.streaming.StreamJob (main): Error Launching job :
org.apache.hadoop.yarn.exceptions.ApplicationNotFoundException: Application
with id 'application_id' doesn't exist in RM.
```

Untuk menyelesaikan masalah ini, Anda harus mengonfigurasi VPC yang menyertakan Set Opsi DHCP dengan parameter yang ditetapkan ke nilai berikut:

> **Note**
>
> Jika Anda menggunakan wilayah AWS GovCloud (AS-barat), tetapkan nama domain menjadi **`us-gov-west-1.compute.internal`** bukan nilai yang digunakan dalam contoh berikut.

- nama domain = **`ec2.internal`**

  Gunakan **`ec2.internal`** jika wilayah Anda adalah US East (N. Virginia). Untuk wilayah lain, gunakan *`region-name`***`.compute.internal`**. Misalnya di us-west-2, gunakan domain-name=**`us-west-2.compute.internal`**.
- domain-name-servers = **`AmazonProvidedDNS`**

Untuk informasi selengkapnya, lihat Set Opsi DHCP.

Kesalahan izin

Kegagalan dalam log `stderr` untuk langkah menunjukkan bahwa sumber daya Amazon S3 tidak memiliki izin yang sesuai. Ini adalah kesalahan 403 dan kesalahannya terlihat seperti:

```
Exception in thread "main" com.amazonaws.services.s3.model.AmazonS3Exception: Access
 Denied (Service: Amazon S3; Status Code: 403; Error Code: AccessDenied; Request
 ID: REQUEST_ID
```

Jika ActionOnFailure disetel ke`TERMINATE_JOB_FLOW`, maka ini akan mengakibatkan cluster berakhir dengan status,`SHUTDOWN_COMPLETED_WITH_ERRORS`.

Beberapa cara untuk memecahkan masalah ini meliputi:

- Jika Anda menggunakan kebijakan bucket Amazon S3 dalam VPC, pastikan untuk memberikan akses ke seluruh bucket dengan membuat VPC endpoint dan memilih Izinkan semua di bawah opsi Kebijakan saat membuat titik akhir.
- Pastikan bahwa setiap kebijakan yang terkait dengan sumber daya S3 menyertakan VPC tempat Anda meluncurkan klaster.
- Coba jalankan perintah berikut ini dari klaster Anda untuk memverifikasi Anda dapat mengakses bucket

  ```
  hadoop fs -copyToLocal s3://path-to-bucket /tmp/
  ```

- Anda dapat mengetahui informasi debugging yang lebih spesifik dengan mengatur parameter `log4j.logger.org.apache.http.wire` ke `DEBUG` dalam file `/home/hadoop/conf/log4j.properties` pada klaster. Anda dapat memeriksa berkas log `stderr` setelah mencoba untuk mengakses bucket dari klaster. Berkas log tersebut akan memberikan informasi lebih detail:

  ```
  Access denied for getting the prefix for bucket - us-west-2.elasticmapreduce with
   path samples/wordcount/input/
  15/03/25 23:46:20 DEBUG http.wire: >> "GET /?prefix=samples%2Fwordcount%2Finput
  %2F&delimiter=%2F&max-keys=1 HTTP/1.1[\r][\n]"
  15/03/25 23:46:20 DEBUG http.wire: >> "Host: us-
  west-2.elasticmapreduce.s3.amazonaws.com[\r][\n]"
  ```

Kesalahan yang mengakibatkan **START_FAILED**

Sebelum AMI 3.7.0, untuk VPC tempat nama host ditentukan, Amazon EMR memetakan nama host internal dari subnet dengan alamat domain kustom sebagai berikut: `ip-X.X.X.X.customdomain.com.tld`. Sebagai contoh, jika nama host itu `ip-10.0.0.10` dan VPC memiliki opsi nama domain yang diatur ke customdomain.com, nama host hasil yang dipetakan oleh Amazon EMR akan `ip-10.0.1.0.customdomain.com`. Sebuah entri ditambahkan dalam `/etc/hosts` untuk menyelesaikan nama host menjadi 10.0.0.10. Perilaku ini diubah dengan AMI 3.7.0 dan sekarang Amazon EMR menghormati konfigurasi DHCP VPC sepenuhnya. Sebelumnya, pelanggan juga dapat menggunakan tindakan bootstrap untuk menentukan pemetaan nama host.

Jika Anda ingin mempertahankan perilaku ini, Anda harus memberikan pengaturan DNS dan resolusi penerusan yang Anda perlukan untuk domain kustom.

Cluster **Terminated with errors** dan NameNode gagal memulai

Ketika meluncurkan klaster EMR di VPC yang menggunakan nama domain DNS kustom, klaster Anda mungkin gagal dengan pesan kesalahan berikut di konsol:

```
Terminated with errors  On the master instance(instance-id), bootstrap action 1
 returned a  non-zero return code
```

Kegagalan adalah akibat dari NameNode tidak bisa memulai. Ini akan menghasilkan kesalahan berikut yang ditemukan di NameNode log, yang URI Amazon S3-nya berbentuk:: `s3://mybucket/logs/cluster-id/daemons/master instance-id/hadoop-hadoop-namenode-master node hostname.log.gz`

```
2015-07-23 20:17:06,266 WARN
      org.apache.hadoop.hdfs.server.namenode.FSNamesystem (main): Encountered
 exception
      loading fsimage  java.io.IOException: NameNode is not formatted.
      at

 org.apache.hadoop.hdfs.server.namenode.FSImage.recoverTransitionRead(FSImage.java:212)
          at

 org.apache.hadoop.hdfs.server.namenode.FSNamesystem.loadFSImage(FSNamesystem.java:1020)
          at

 org.apache.hadoop.hdfs.server.namenode.FSNamesystem.loadFromDisk(FSNamesystem.java:739)
```

```
        at
    org.apache.hadoop.hdfs.server.namenode.NameNode.loadNamesystem(NameNode.java:537)
        at
    org.apache.hadoop.hdfs.server.namenode.NameNode.initialize(NameNode.java:596)

    at  org.apache.hadoop.hdfs.server.namenode.NameNode.<init>(NameNode.java:765)
        at
    org.apache.hadoop.hdfs.server.namenode.NameNode.<init>(NameNode.java:749)
    at

org.apache.hadoop.hdfs.server.namenode.NameNode.createNameNode(NameNode.java:1441)
        at
    org.apache.hadoop.hdfs.server.namenode.NameNode.main(NameNode.java:1507)
```

Hal ini disebabkan potensi masalah ketika instans EC2 dapat memiliki beberapa set nama domain yang memenuhi syarat saat meluncurkan klaster EMR di VPC, yang menggunakan server DNS yang disediakan oleh AWS dan server DNS kustom yang disediakan pengguna. Jika server DNS yang disediakan pengguna tidak menyediakan catatan pointer (PTR) untuk setiap catatan A yang digunakan untuk menunjuk simpul dalam klaster EMR, klaster akan gagal memulai ketika dikonfigurasi dengan cara ini. Solusinya adalah dengan menambahkan 1 catatan PTR untuk setiap catatan A yang dibuat ketika instans EC2 diluncurkan di salah satu subnet di VPC.

Kesalahan klaster streaming

Anda biasanya dapat menemukan penyebab kesalahan streaming di file `syslog`. Tautkan ke sana pada panel Langkah.

Kesalahan berikut umum terjadi untuk klaster streaming.

Topik

- Apakah data yang dikirim ke pemeta memiliki format yang salah?
- Apakah skrip Anda kehabisan waktu?
- Apakah Anda meloloskan argumen streaming yang tidak valid?
- Apakah skrip Anda keluar dengan kesalahan?

Apakah data yang dikirim ke pemeta memiliki format yang salah?

Untuk memeriksa apakah hal ini yang terjadi, cari pesan kesalahan di file `syslog` pada upaya tugas yang gagal di log upaya tugas. Untuk informasi selengkapnya, lihat Melihat berkas log.

Apakah skrip Anda kehabisan waktu?

Waktu habis default untuk skrip pemeta atau peredam adalah 600 detik. Jika skrip Anda membutuhkan waktu lebih lama, upaya tugas akan gagal. Anda dapat memverifikasi apakah hal ini yang terjadi dengan memeriksa file `syslog` pada upaya tugas yang gagal di log upaya tugas. Untuk informasi selengkapnya, lihat Melihat berkas log.

Anda dapat mengubah batas waktu dengan menetapkan nilai baru untuk pengaturan konfigurasi `mapred.task.timeout`. Pengaturan ini menentukan jumlah milidetik sebelum Amazon EMR akan mengakhiri tugas yang belum membaca input, menuliskan output, atau memperbarui string statusnya. Anda dapat memperbarui nilai ini dengan meloloskan argumen streaming tambahan `-jobconf mapred.task.timeout=800000`.

Apakah Anda meloloskan argumen streaming yang tidak valid?

Hadoop streaming hanya mendukung argumen berikut. Jika Anda meloloskan argumen selain yang tercantum di bawah ini, klaster akan gagal.

```
-blockAutoGenerateCacheFiles
-cacheArchive
-cacheFile
-cmdenv
-combiner
-debug
-input
-inputformat
-inputreader
-jobconf
-mapper
-numReduceTasks
-output
-outputformat
-partitioner
-reducer
-verbose
```

Selain itu, Hadoop streaming hanya mengenali argumen yang diloloskan menggunakan sintaks Java; yaitu, didahului oleh tanda hubung tunggal. Jika Anda meloloskan argumen yang didahului oleh tanda hubung ganda, klaster akan gagal.

Apakah skrip Anda keluar dengan kesalahan?

Jika skrip pemeta atau peredam Anda keluar dengan kesalahan, Anda dapat menemukan kesalahan dalam file `stderr` pada log upaya tugas dari upaya tugas yang gagal. Untuk informasi selengkapnya, lihat Melihat berkas log.

Kesalahan klaster JAR kustom

Kesalahan berikut umum terjadi untuk klaster JAR kustom.

Topik

- Apakah JAR Anda membuang pengecualian sebelum membuat pekerjaan?
- Apakah JAR Anda membuang kesalahan di dalam tugas pemetaan?

Apakah JAR Anda membuang pengecualian sebelum membuat pekerjaan?

Jika program utama JAR kustom Anda membuang pengecualian saat membuat pekerjaan Hadoop, tempat terbaik untuk mencarinya adalah file `syslog` log langkah. Untuk informasi selengkapnya, lihat Melihat berkas log.

Apakah JAR Anda membuang kesalahan di dalam tugas pemetaan?

Jika JAR kustom dan pemeta Anda membuang pengecualian saat memproses data input, tempat terbaik untuk mencarinya adalah file `syslog` log upaya tugas. Untuk informasi selengkapnya, lihat Melihat berkas log.

AWS GovCloud Kesalahan (AS-Barat)

Wilayah AWS GovCloud (AS-Barat) berbeda dari wilayah lain dalam keamanan, konfigurasi, dan pengaturan defaultnya. Akibatnya, gunakan daftar periksa berikut untuk memecahkan masalah kesalahan EMR Amazon yang spesifik untuk wilayah (AS-Barat) sebelum menggunakan rekomendasi AWS GovCloud pemecahan masalah yang lebih umum.

- Verifikasi bahwa IAM role Anda dikonfigurasi dengan benar. Untuk informasi selengkapnya, lihat Konfigurasi peran layanan IAM untuk izin Amazon EMR untuk layanan AWS dan sumber daya.
- Pastikan bahwa konfigurasi VPC Anda telah mengonfigurasi parameter dukungan resplusi DNS/ nama host, Gateway Internet, dan Set Opsi DHCP dengan benar. Untuk informasi selengkapnya, lihat Kesalahan VPC.

Jika langkah-langkah ini tidak memecahkan masalah, lanjutkan dengan langkah-langkah untuk memecahkan kesalahan umum Amazon EMR. Untuk informasi selengkapnya, lihat Kesalahan umum di Amazon EMR.

Temukan cluster yang hilang

Jika klaster Anda hilang dari daftar konsol atau `ListClusters` API, periksa hal berikut:

- Konfirmasikan bahwa usia cluster dari saat penyelesaian kurang dari dua bulan. Amazon EMR menyimpan informasi metadata untuk cluster yang telah selesai selama dua bulan tanpa biaya. Anda tidak dapat menghapus cluster yang sudah selesai dari konsol — sebagai gantinya, Amazon EMR membersihkan cluster yang sudah selesai secara otomatis setelah dua bulan.
- Konfirmasikan bahwa Anda memiliki izin peran untuk melihat klaster.
- Konfirmasikan bahwa Anda melihat hal yang sama Wilayah AWS di mana cluster berada.

Memecahkan masalah klaster gagal

Bagian ini memandu Anda melalui proses pemecahan masalah klaster yang telah gagal. Ini berarti bahwa klaster diakhiri dengan kode kesalahan.

> **Note**
>
> Ketika kluster EMR berakhir dengan kesalahan, `ListClusters` API `DescribeCluster` dan mengembalikan kode kesalahan dan pesan kesalahan. Untuk beberapa kesalahan cluster, array `ErrorDetail` data juga dapat membantu Anda memecahkan masalah kegagalan. Untuk informasi selengkapnya, lihat Kode kesalahan dengan ErrorDetail informasi.

Jika cluster Anda berjalan tetapi membutuhkan waktu lama untuk mengembalikan hasil, lihatMemecahkan masalah klaster lambat.

Topik

- Langkah 1: Kumpulkan data tentang masalah
- Langkah 2: Periksa lingkungan
- Langkah 3: Periksa perubahan status terakhir

Langkah 1: Kumpulkan data tentang masalah

Langkah pertama dalam pemecahan masalah klaster adalah mengumpulkan informasi tentang apa yang salah serta status dan konfigurasi klaster saat ini. Informasi ini akan digunakan dalam langkah-langkah berikut untuk mengkonfirmasi atau mengesampingkan kemungkinan penyebab masalah.

Menentukan masalah

Definisi yang jelas tentang masalah yang terjadi adalah hal pertama yang dilakukan. Beberapa pertanyaan untuk Anda tanyakan pada diri sendiri:

- Apa yang saya harapkan terjadi? Apa yang terjadi sebagai gantinya?
- Kapan masalah ini pertama kali terjadi? Seberapa sering hal itu terjadi sejak pertama kali ditemukan?
- Apakah ada yang berubah dalam cara saya mengonfigurasi atau menjalankan klaster saya?

Detail klaster

Detail klaster berikut berguna dalam membantu melacak masalah. Untuk informasi selengkapnya tentang cara mengumpulkan informasi ini, lihat Melihat status dan detail klaster.

- Pengidentifikasi klaster. (Juga disebut pengenal alur kerja.)
- Wilayah AWSdan Availability Zone tempat cluster diluncurkan.
- Status klaster, termasuk detail perubahan status terakhir.
- Jenis dan jumlah instans EC2 yang ditentukan untuk simpul utama, inti, dan tugas.

Langkah 2: Periksa lingkungan

Amazon EMR beroperasi sebagai bagian dari ekosistem layanan web dan perangkat lunak sumber terbuka. Hal-hal yang mempengaruhi dependensi tersebut dapat mempengaruhi performa Amazon EMR.

Topik

- Periksa pemadaman layanan
- Periksa batas penggunaan
- Periksa versi rilis
- Periksa konfigurasi subnet Amazon VPC

Periksa pemadaman layanan

Amazon EMR menggunakan beberapa Amazon Web Services secara internal. Ini menjalankan server virtual di Amazon EC2, menyimpan data dan skrip di Amazon S3, dan melaporkan metrik ke. CloudWatch Peristiwa yang mengganggu layanan ini jarang terjadi — tetapi ketika terjadi — dapat menyebabkan masalah di Amazon EMR.

Sebelum Anda melangkah lebih jauh, periksa Service Health Dashboard. Periksa Wilayah tempat Anda meluncurkan klaster untuk melihat apakah ada peristiwa gangguan di salah satu layanan ini.

Periksa batas penggunaan

Jika Anda meluncurkan cluster besar, telah meluncurkan banyak cluster secara bersamaan, atau Anda adalah pengguna yang berbagi Akun AWS dengan pengguna lain, cluster mungkin gagal karena Anda melebihi batas AWS layanan.

Amazon EC2 membatasi jumlah instans server virtual yang berjalan di satu AWS Wilayah hingga 20 instans sesuai permintaan atau cadangan. Jika Anda meluncurkan cluster dengan lebih dari 20 node, atau meluncurkan cluster yang menyebabkan jumlah total instans EC2 yang aktif pada Anda Akun AWS melebihi 20, cluster tidak akan dapat meluncurkan semua instans EC2 yang diperlukan dan mungkin gagal. Ketika ini terjadi, Amazon EMR mengembalikan kesalahan `EC2 QUOTA EXCEEDED`. Anda dapat meminta bahwa AWS meningkatkan jumlah instans EC2 yang dapat Anda jalankan di akun Anda dengan mengirimkan aplikasi Permintaan untuk Meningkatkan Batas Instans Amazon EC2.

Hal lain yang dapat menyebabkan Anda melebihi batas penggunaan Anda adalah penundaan antara ketika sebuah klaster diakhiri dan ketika klaster merilis seluruh sumber dayanya. Tergantung pada konfigurasinya, mungkin memakan waktu hingga 5-20 menit bagi klaster untuk sepenuhnya mengakhiri dan melepaskan sumber daya yang teralokasi. Jika Anda mendapatkan kesalahan `EC2 QUOTA EXCEEDED` ketika Anda mencoba untuk memulai sebuah klaster, ini mungkin karena sumber daya dari klaster yang baru diakhiri belum dirilis. Dalam kasus ini, Anda dapat meminta kuota Amazon EC2 Anda ditingkatkan, atau Anda dapat menunggu dua puluh menit dan meluncurkan ulang klaster tersebut.

Amazon S3 membatasi jumlah bucket yang dibuat pada sebuah akun hingga 100. Jika klaster Anda menciptakan bucket baru yang melebihi batas ini, pembuatan bucket akan gagal dan dapat menyebabkan klaster gagal.

Periksa versi rilis

Bandingkan label rilis yang Anda gunakan untuk meluncurkan klaster dengan rilis Amazon EMR terbaru. Setiap rilis Amazon EMR mencakup perbaikan seperti aplikasi, fitur, patch, dan perbaikan bug yang baru. Masalah yang mempengaruhi klaster Anda mungkin telah diperbaiki dalam versi rilis terbaru. Jika memungkinkan, jalankan kembali klaster Anda menggunakan versi terbaru.

Periksa konfigurasi subnet Amazon VPC

Jika klaster Anda diluncurkan di subnet Amazon VPC, subnet harus dikonfigurasi seperti yang dijelaskan di Mengkonfigurasi jaringan. Selain itu, periksa bahwa subnet tempat Anda meluncurkan klaster memiliki alamat IP elastis kosong yang cukup untuk menugaskan satu untuk setiap simpul dalam klaster.

Langkah 3: Periksa perubahan status terakhir

Perubahan status terakhir memberikan informasi tentang apa yang terjadi terakhir kali status klaster berubah. Hal ini sering memiliki informasi yang dapat memberitahu Anda apa yang salah ketika klaster berubah status menjadi `FAILED`. Sebagai contoh, jika Anda meluncurkan klaster streaming dan menentukan lokasi output yang sudah ada di Amazon S3, klaster akan gagal dengan perubahan status terakhir berbunyi “Direktori output streaming sudah ada”.

Anda dapat menemukan nilai perubahan status terakhir dari konsol dengan melihat panel detail untuk klaster, dari CLI menggunakan argumen `list-steps` atau `describe-cluster`, atau dari API menggunakan tindakan `DescribeCluster` dan `ListSteps`. Untuk informasi selengkapnya, lihat Melihat status dan detail klaster.

Langkah 4: Memeriksa berkas log

Langkah berikutnya adalah memeriksa berkas log untuk menemukan kode kesalahan atau indikasi lain dari masalah yang dialami klaster Anda. Untuk informasi tentang berkas log yang tersedia, tempat menemukannya, dan bagaimana melihatnya, lihat Melihat berkas log.

Mungkin diperlukan beberapa pekerjaan investigasi untuk menentukan apa yang terjadi. Hadoop menjalankan pekerjaan dalam upaya tugas pada berbagai simpul dalam klaster. Amazon EMR dapat

memulai upaya tugas spekulatif, mengakhiri upaya tugas lain yang tidak selesai terlebih dahulu. Hal ini menghasilkan aktivitas yang signifikan yang di-log ke berkas log pengendali, stderr dan syslog saat terjadi. Selain itu, beberapa upaya tugas berjalan secara bersamaan, tetapi berkas log hanya dapat menampilkan hasil secara linier.

Mulailah dengan memeriksa log tindakan bootstrap untuk mengetahui kesalahan atau perubahan konfigurasi yang tidak terduga selama peluncuran klaster. Dari sana, lihat di log langkah untuk mengidentifikasi pekerjaan Hadoop yang diluncurkan sebagai bagian dari langkah dengan kesalahan. Periksa log pekerjaan Hadoop untuk mengidentifikasi upaya tugas yang gagal. Log upaya tugas akan berisi detail tentang apa yang menyebabkan suatu upaya tugas gagal.

Bagian berikut ini menjelaskan cara menggunakan berbagai berkas log untuk mengidentifikasi kesalahan dalam klaster Anda.

Periksa log tindakan bootstrap

Tindakan bootstrap menjalankan skrip pada klaster saat klaster diluncurkan. Mereka biasanya digunakan untuk menginstal perangkat lunak tambahan pada klaster atau untuk mengubah pengaturan konfigurasi dari nilai default. Memeriksa log ini dapat memberikan wawasan tentang kesalahan yang terjadi selama mengatur klaster serta perubahan pengaturan konfigurasi yang dapat mempengaruhi performa.

Periksa log langkah

Ada empat jenis log langkah.

- pengendali -Berisi file yang dihasilkan oleh Amazon EMR (Amazon EMR) yang muncul dari kesalahan yang dihadapi ketika mencoba untuk menjalankan langkah Anda. Jika langkah Anda gagal saat memuat, Anda dapat menemukan jejak tumpukan dalam log ini. Kesalahan memuat atau mengakses aplikasi Anda seringkali dijelaskan di sini, seperti kesalahan file pemeta hilang.
- stderr—Berisi pesan kesalahan yang terjadi saat memproses langkah. Kesalahan memuat aplikasi sering kali dijelaskan di sini. Log ini kadang-kadang berisi jejak tumpukan.
- stdout -Berisi status yang dihasilkan oleh pemeta dan peredam yang dapat dieksekusi. Kesalahan memuat aplikasi sering kali dijelaskan di sini. Log ini kadang-kadang berisi pesan kesalahan aplikasi.
- syslog—Berisi log dari perangkat lunak non-Amazon, seperti Apache dan Hadoop. Kesalahan streaming seringkali dijelaskan di sini.

Periksa stderr untuk kesalahan yang jelas. Jika stderr menampilkan daftar singkat kesalahan, langkah akan segera berhenti dengan kesalahan yang terjadi. Hal ini paling sering disebabkan oleh kesalahan dalam aplikasi pemeta dan peredam yang dijalankan di klaster.

Periksa baris terakhir dari pengendali dan syslog untuk melihat pemberitahuan kesalahan atau kegagalan. Ikuti pemberitahuan tentang tugas yang gagal, terutama jika tertulis “Pekerjaan Gagal”.

Periksa log upaya tugas

Jika analisis sebelumnya dari log langkah menimbulkan satu tugas yang gagal atau lebih, selidiki log dari upaya tugas yang sesuai untuk melihat informasi kesalahan yang lebih detail.

Langkah 5: Uji klaster langkah demi langkah

Teknik yang berguna ketika Anda mencoba untuk melacak sumber kesalahan adalah memulai ulang klaster dan mengirimkan langkah-langkah ke klaster tersebut satu per satu. Hal ini memungkinkan Anda memeriksa hasil dari setiap langkah sebelum memproses langkah berikutnya, dan memberi Anda kesempatan untuk memperbaiki dan menjalankan kembali langkah yang telah gagal. Keuntungan lain adalah Anda hanya memuat data input Anda satu kali.

Untuk menguji langkah klaster langkah demi langkah

1. Luncurkan klaster baru, dengan keep alive dan proteksi pengakhiran diaktifkan. Keep alive membuat klaster tetap berjalan setelah klaster memproses semua langkah-langkah yang tertunda. Protensi pengakhiran mencegah klaster untuk mati ketika terjadi kesalahan. Untuk informasi selengkapnya, lihat Mengkonfigurasi cluster untuk melanjutkan atau mengakhiri setelah eksekusi langkah dan Menggunakan perlindungan pengakhiran.
2. Kirim langkah ke klaster. Untuk informasi selengkapnya, lihat Kirim pekerjaan ke sebuah klaster.
3. Setelah langkah selesai memproses, periksa kesalahan dalam berkas log langkah. Untuk informasi selengkapnya, lihat Langkah 4: Memeriksa berkas log. Cara tercepat untuk menemukan berkas log ini adalah dengan menghubungkan ke simpul utama dan melihat berkas log di sana. Berkas log langkah tidak muncul sampai langkah berjalan untuk beberapa waktu, selesai, atau gagal.
4. Jika langkah berhasil tanpa kesalahan, jalankan langkah berikutnya. Jika terjadi kesalahan, selidiki kesalahan dalam berkas log. Jika kesalahan ada pada kode Anda, lakukan koreksi dan jalankan ulang langkah. Lanjutkan sampai semua langkah berjalan tanpa kesalahan.

5. Ketika Anda selesai debugging klaster, dan ingin mengakhiri klaster, Anda harus mengakhirinya secara manual. Hal ini diperlukan karena klaster diluncurkan dengan proteksi pengakhiran yang diaktifkan. Untuk informasi selengkapnya, lihat Menggunakan perlindungan pengakhiran.

Memecahkan masalah klaster lambat

Bagian ini memandu Anda melalui proses pemecahan masalah klaster yang masih berjalan, tetapi membutuhkan waktu lama untuk mengembalikan hasil. Untuk informasi selengkapnya tentang apa yang harus dilakukan jika klaster diakhiri dengan kode kesalahan, lihat Memecahkan masalah klaster gagal

Amazon EMR mengizinkan Anda untuk menentukan jumlah dan jenis instans dalam klaster. Spesifikasi ini adalah sarana utama untuk memengaruhi kecepatan untuk menyelesaikan pemrosesan data Anda. Satu hal yang mungkin Anda pertimbangkan adalah menjalankan ulang klaster, kali ini menentukan instans EC2 dengan sumber daya yang lebih besar, atau menentukan jumlah instans yang lebih besar dalam klaster. Untuk informasi selengkapnya, lihat Konfigurasi perangkat keras dan jaringan klaster.

Topik berikut ini memandu Anda dalam proses mengidentifikasi penyebab alternatif klaster lambat.

Topik

Langkah 1: Kumpulkan data tentang masalah

Langkah pertama dalam pemecahan masalah klaster adalah mengumpulkan informasi tentang apa yang salah serta status dan konfigurasi klaster saat ini. Informasi ini akan digunakan dalam langkah-langkah berikut untuk mengkonfirmasi atau mengesampingkan kemungkinan penyebab masalah.

Menentukan masalah

Definisi yang jelas tentang masalah yang terjadi adalah hal pertama yang dilakukan. Beberapa pertanyaan untuk Anda tanyakan pada diri sendiri:

- Apa yang saya harapkan terjadi? Apa yang terjadi sebagai gantinya?
- Kapan masalah ini pertama kali terjadi? Seberapa sering hal itu terjadi sejak pertama kali ditemukan?
- Apakah ada yang berubah dalam cara saya mengonfigurasi atau menjalankan klaster saya?

Detail klaster

Detail klaster berikut berguna dalam membantu melacak masalah. Untuk informasi selengkapnya tentang cara mengumpulkan informasi ini, lihat Melihat status dan detail klaster.

- Pengidentifikasi klaster. (Juga disebut pengenal alur kerja.)
- Wilayah AWSdan Availability Zone tempat cluster diluncurkan.
- Status klaster, termasuk detail perubahan status terakhir.
- Jenis dan jumlah instans EC2 yang ditentukan untuk simpul utama, inti, dan tugas.

Langkah 2: Periksa lingkungan

Topik

- Periksa pemadaman layanan
- Periksa batas penggunaan
- Periksa konfigurasi subnet Amazon VPC
- Memulai ulang klaster

Periksa pemadaman layanan

Amazon EMR menggunakan beberapa Amazon Web Services secara internal. Ini menjalankan server virtual di Amazon EC2, menyimpan data dan skrip di Amazon S3, dan melaporkan metrik ke. CloudWatch Peristiwa yang mengganggu layanan ini jarang terjadi — tetapi ketika terjadi — dapat menyebabkan masalah di Amazon EMR.

Sebelum Anda melangkah lebih jauh, periksa Service Health Dashboard. Periksa Wilayah tempat Anda meluncurkan klaster untuk melihat apakah ada peristiwa gangguan di salah satu layanan ini.

Periksa batas penggunaan

Jika Anda meluncurkan cluster besar, telah meluncurkan banyak cluster secara bersamaan, atau Anda adalah pengguna yang berbagi Akun AWS dengan pengguna lain, cluster mungkin gagal karena Anda melebihi batas AWS layanan.

Amazon EC2 membatasi jumlah instans server virtual yang berjalan di satu AWS Wilayah hingga 20 instans sesuai permintaan atau cadangan. Jika Anda meluncurkan cluster dengan lebih dari 20 node, atau meluncurkan cluster yang menyebabkan jumlah total instans EC2 yang aktif pada Anda Akun AWS melebihi 20, cluster tidak akan dapat meluncurkan semua instans EC2 yang diperlukan dan mungkin gagal. Ketika ini terjadi, Amazon EMR mengembalikan kesalahan `EC2 QUOTA EXCEEDED`. Anda dapat meminta bahwa AWS meningkatkan jumlah instans EC2 yang dapat Anda jalankan di akun Anda dengan mengirimkan aplikasi Permintaan untuk Meningkatkan Batas Instans Amazon EC2.

Hal lain yang dapat menyebabkan Anda melebihi batas penggunaan Anda adalah penundaan antara ketika sebuah klaster diakhiri dan ketika klaster merilis seluruh sumber dayanya. Tergantung pada konfigurasinya, mungkin memakan waktu hingga 5-20 menit bagi klaster untuk sepenuhnya mengakhiri dan melepaskan sumber daya yang teralokasi. Jika Anda mendapatkan kesalahan `EC2 QUOTA EXCEEDED` ketika Anda mencoba untuk memulai sebuah klaster, ini mungkin karena sumber daya dari klaster yang baru diakhiri belum dirilis. Dalam kasus ini, Anda dapat meminta kuota Amazon EC2 Anda ditingkatkan, atau Anda dapat menunggu dua puluh menit dan meluncurkan ulang klaster tersebut.

Amazon S3 membatasi jumlah bucket yang dibuat pada sebuah akun hingga 100. Jika klaster Anda menciptakan bucket baru yang melebihi batas ini, pembuatan bucket akan gagal dan dapat menyebabkan klaster gagal.

Periksa konfigurasi subnet Amazon VPC

Jika klaster Anda diluncurkan di subnet Amazon VPC, subnet harus dikonfigurasi seperti yang dijelaskan di Mengkonfigurasi jaringan. Selain itu, periksa bahwa subnet tempat Anda meluncurkan klaster memiliki alamat IP elastis kosong yang cukup untuk menugaskan satu untuk setiap simpul dalam klaster.

Memulai ulang klaster

Pelambatan dalam pemrosesan mungkin disebabkan oleh kondisi sementara. Pertimbangkan untuk mengakhiri dan memulai ulang klaster untuk melihat apakah performa meningkat.

Langkah 3: Memeriksa berkas log

Langkah berikutnya adalah memeriksa berkas log untuk menemukan kode kesalahan atau indikasi lain dari masalah yang dialami klaster Anda. Untuk informasi tentang berkas log yang tersedia, tempat menemukannya, dan bagaimana melihatnya, lihat Melihat berkas log.

Mungkin diperlukan beberapa pekerjaan investigasi untuk menentukan apa yang terjadi. Hadoop menjalankan pekerjaan dalam upaya tugas pada berbagai simpul dalam klaster. Amazon EMR dapat memulai upaya tugas spekulatif, mengakhiri upaya tugas lain yang tidak selesai terlebih dahulu. Hal ini menghasilkan aktivitas yang signifikan yang di-log ke berkas log pengendali, stderr dan syslog saat terjadi. Selain itu, beberapa upaya tugas berjalan secara bersamaan, tetapi berkas log hanya dapat menampilkan hasil secara linier.

Mulailah dengan memeriksa log tindakan bootstrap untuk mengetahui kesalahan atau perubahan konfigurasi yang tidak terduga selama peluncuran klaster. Dari sana, lihat di log langkah untuk mengidentifikasi pekerjaan Hadoop yang diluncurkan sebagai bagian dari langkah dengan kesalahan. Periksa log pekerjaan Hadoop untuk mengidentifikasi upaya tugas yang gagal. Log upaya tugas akan berisi detail tentang apa yang menyebabkan suatu upaya tugas gagal.

Bagian berikut ini menjelaskan cara menggunakan berbagai berkas log untuk mengidentifikasi kesalahan dalam klaster Anda.

Periksa log tindakan bootstrap

Tindakan bootstrap menjalankan skrip pada klaster saat klaster diluncurkan. Mereka biasanya digunakan untuk menginstal perangkat lunak tambahan pada klaster atau untuk mengubah pengaturan konfigurasi dari nilai default. Memeriksa log ini dapat memberikan wawasan tentang kesalahan yang terjadi selama mengatur klaster serta perubahan pengaturan konfigurasi yang dapat mempengaruhi performa.

Periksa log langkah

Ada empat jenis log langkah.

- pengendali -Berisi file yang dihasilkan oleh Amazon EMR (Amazon EMR) yang muncul dari kesalahan yang dihadapi ketika mencoba untuk menjalankan langkah Anda. Jika langkah Anda

gagal saat memuat, Anda dapat menemukan jejak tumpukan dalam log ini. Kesalahan memuat atau mengakses aplikasi Anda seringkali dijelaskan di sini, seperti kesalahan file pemeta hilang.

- stderr—Berisi pesan kesalahan yang terjadi saat memproses langkah. Kesalahan memuat aplikasi sering kali dijelaskan di sini. Log ini kadang-kadang berisi jejak tumpukan.
- stdout -Berisi status yang dihasilkan oleh pemeta dan peredam yang dapat dieksekusi. Kesalahan memuat aplikasi sering kali dijelaskan di sini. Log ini kadang-kadang berisi pesan kesalahan aplikasi.
- syslog—Berisi log dari perangkat lunak non-Amazon, seperti Apache dan Hadoop. Kesalahan streaming seringkali dijelaskan di sini.

Periksa stderr untuk kesalahan yang jelas. Jika stderr menampilkan daftar singkat kesalahan, langkah akan segera berhenti dengan kesalahan yang terjadi. Hal ini paling sering disebabkan oleh kesalahan dalam aplikasi pemeta dan peredam yang dijalankan di klaster.

Periksa baris terakhir dari pengendali dan syslog untuk melihat pemberitahuan kesalahan atau kegagalan. Ikuti pemberitahuan tentang tugas yang gagal, terutama jika tertulis “Pekerjaan Gagal”.

Periksa log upaya tugas

Jika analisis sebelumnya dari log langkah menimbulkan satu tugas yang gagal atau lebih, selidiki log dari upaya tugas yang sesuai untuk melihat informasi kesalahan yang lebih detail.

Periksa log daemon Hadoop

Dalam kasus yang jarang terjadi, Hadoop sendiri mungkin gagal. Untuk melihat apakah itu yang terjadi, Anda harus melihat log Hadoop. Log ini ada di `/var/log/hadoop/` pada setiap simpul.

Anda dapat menggunakan JobTracker log untuk memetakan upaya tugas yang gagal ke simpul yang dijalankan. Setelah Anda mengetahui simpul yang terkait dengan upaya tugas tersebut, Anda dapat memeriksa kesehatan instans EC2 yang menjadi host simpul tersebut untuk melihat apakah ada masalah seperti kehabisan CPU atau memori.

Langkah 4: Periksa kesehatan klaster dan instans

Sebuah klaster Amazon EMR terdiri dari simpul yang berjalan di instans Amazon EC2. Jika instans tersebut terikat sumber daya (seperti kehabisan CPU atau memori), mengalami masalah konektivitas jaringan, atau diakhiri, kecepatan pemrosesan klaster akan terganggu.

Ada hingga tiga jenis simpul dalam sebuah klaster:

- Simpul Utama — mengelola klaster. Jika mengalami masalah performa, seluruh klaster terpengaruh.
- Simpul Inti — memproses tugas pemetaan-peredam dan memelihara Hadoop Distributed Filesystem (HDFS). Jika salah satu simpul ini mengalami masalah performa, hal itu dapat memperlambat operasi HDFS serta pemrosesan pemetaan-peredaman. Anda dapat menambahkan simpul inti tambahan ke suatu klaster untuk meningkatkan performa, tetapi tidak dapat menghapus simpul inti. Untuk informasi selengkapnya, lihat Secara manual mengubah ukuran klaster berjalan.
- simpul tugas — memproses tugas pemetaan-peredaman. Simpul ini adalah sumber komputasi murni dan tidak menyimpan data. Anda dapat menambahkan simpul tugas ke sebuah klaster untuk mempercepat performa, atau menghapus simpul tugas yang tidak diperlukan. Untuk informasi selengkapnya, lihat Secara manual mengubah ukuran klaster berjalan.

Ketika Anda melihat kesehatan klaster, Anda harus melihat performa klaster secara keseluruhan, serta performa masing-masing instans. Ada beberapa alat yang dapat Anda gunakan:

Periksa kesehatan cluster dengan CloudWatch

Setiap klaster EMR Amazon melaporkan metrik ke. CloudWatch Metrik ini memberikan ringkasan informasi performa tentang klaster, seperti total beban, pemanfaatan HDFS, tugas berjalan, tugas yang tersisa, blok yang rusak, dan banyak lagi. Melihat CloudWatch metrik memberi Anda gambaran besar tentang apa yang terjadi dengan cluster Anda dan dapat memberikan wawasan tentang apa yang menyebabkan perlambatan dalam pemrosesan. Selain menggunakan CloudWatch untuk menganalisis masalah kinerja yang ada, Anda dapat menyetel alarm yang CloudWatch menyebabkan peringatan jika terjadi masalah kinerja di masa mendatang. Untuk informasi selengkapnya, lihat Memantau metrik Amazon EMR dengan CloudWatch.

Periksa status pekerjaan dan kesehatan HDFS

Gunakan tab Antarmuka pengguna aplikasi pada halaman detail klaster untuk melihat detail aplikasi YARN. Untuk aplikasi tertentu, Anda dapat menelusuri detail lebih lanjut dan mengakses log secara langsung. Hal ini sangat berguna untuk aplikasi Spark. Untuk informasi selengkapnya, lihat Melihat riwayat aplikasi.

Hadoop menyediakan serangkaian antarmuka web yang dapat Anda gunakan untuk melihat informasi. Untuk informasi selengkapnya tentang cara mengakses antarmuka web ini, lihat Melihat antarmuka web yang di-host pada klaster Amazon EMR.

- JobTracker — memberikan informasi tentang kemajuan pekerjaan yang sedang diproses oleh cluster. Anda dapat menggunakan antarmuka ini untuk mengidentifikasi kapan pekerjaan menjadi macet.
- HDFS NameNode — memberikan informasi tentang persentase pemanfaatan HDFS dan ruang yang tersedia pada setiap node. Anda dapat menggunakan antarmuka ini untuk mengidentifikasi ketika HDFS menjadi terikat sumber daya dan membutuhkan kapasitas tambahan.
- TaskTracker — memberikan informasi tentang tugas-tugas pekerjaan yang sedang diproses oleh cluster. Anda dapat menggunakan antarmuka ini untuk mengidentifikasi kapan tugas menjadi macet.

Periksa kesehatan instans dengan Amazon EC2

Cara lain untuk mencari informasi tentang status instans di klaster Anda adalah dengan menggunakan konsol Amazon EC2. Karena setiap simpul dalam klaster berjalan pada instans EC2, Anda dapat menggunakan alat-alat yang disediakan oleh Amazon EC2 untuk memeriksa status mereka. Untuk informasi selengkapnya, lihat Melihat instans klaster di Amazon EC2.

Langkah 5: Periksa grup yang ditangguhkan

Grup instans menjadi ditangguhkan ketika ia menemui terlalu banyak kesalahan ketika mencoba untuk meluncurkan simpul. Sebagai contoh, jika simpul baru berulang kali gagal saat melakukan tindakan bootstrap, grup instans akan — setelah beberapa waktu — memasuki status `SUSPENDED` dan tidak terus mencoba untuk menyediakan simpul baru.

Suatu simpul dapat gagal muncul jika:

- Hadoop atau klaster ternyata rusak dan tidak menerima simpul baru ke dalam klaster
- Tindakan bootstrap gagal pada simpul baru
- Simpul tidak berfungsi dengan benar dan gagal untuk check in dengan Hadoop

Jika grup instans berada pada status `SUSPENDED`, dan klaster berada dalam status `WAITING`, Anda dapat menambahkan langkah klaster untuk menyetel ulang jumlah simpul inti dan simpul tugas yang diinginkan. Menambahkan pemrosesan melanjutkan langkah klaster dan menempatkan grup instans kembali ke status `RUNNING`.

Untuk informasi selengkapnya tentang cara menyetel ulang klaster dalam keadaan ditangguhkan, lihat Kondisi yang ditangguhkan.

Langkah 6: Meninjau pengaturan konfigurasi

Pengaturan konfigurasi menentukan detail tentang bagaimana klaster berjalan, seperti berapa kali untuk mencoba kembali tugas dan berapa banyak memori tersedia untuk menyortir. Ketika Anda meluncurkan klaster menggunakan Amazon EMR, ada pengaturan khusus Amazon EMR selain pengaturan konfigurasi Hadoop standar. Pengaturan konfigurasi disimpan pada simpul utama klaster. Anda dapat memeriksa pengaturan konfigurasi untuk memastikan bahwa klaster Anda memiliki sumber daya yang diperlukan untuk berjalan secara efisien.

Amazon EMR mendefinisikan pengaturan konfigurasi default Hadoop yang digunakan untuk meluncurkan klaster. Nilai-nilainya didasarkan pada AMI dan tipe instans yang Anda tentukan untuk klaster. Anda dapat memodifikasi pengaturan konfigurasi ini dari nilai default menggunakan tindakan bootstrap atau dengan menentukan nilai-nilai baru dalam parameter eksekusi pekerjaan. Untuk informasi selengkapnya, lihat Buat tindakan bootstrap untuk menginstal perangkat lunak tambahan. Untuk menentukan apakah tindakan bootstrap mengubah pengaturan konfigurasi, periksa log tindakan bootstrap.

Amazon EMR mencatat pengaturan Hadoop yang digunakan untuk melaksanakan setiap pekerjaan. Data log disimpan dalam sebuah file bernama `job_`*`job-id`*`_conf.xml` di bawah direktori `/mnt/var/log/hadoop/history/` simpul utama, dimana *`job-id`* digantikan oleh pengidentifikasi pekerjaan. Jika Anda telah mengaktifkan pengarsipan log, data ini akan disalin ke Amazon S3 dalam folder `logs/`*`date`*`/`*`jobflow-id`*`/jobs`, dimana *`tanggal`* adalah tanggal pekerjaan berjalan, dan *`jobflow-id`* adalah pengidentifikasi klaster.

Pengaturan konfigurasi pekerjaan Hadoop berikut ini sangat berguna untuk menyelidiki masalah performa. Untuk informasi selengkapnya tentang pengaturan konfigurasi Hadoop dan cara mereka mempengaruhi perilaku Hadoop, buka http://hadoop.apache.org/docs/.

> ⚠ Warning
>
> 1. Pengaturan `dfs.replication` ke 1 pada cluster dengan kurang dari empat node dapat menyebabkan hilangnya data HDFS jika satu node turun. Kami menyarankan Anda menggunakan cluster dengan setidaknya empat node inti untuk beban kerja produksi.
> 2. Amazon EMR tidak akan mengizinkan cluster untuk menskalakan node inti di bawah ini. `dfs.replication` Misalnya, jika`dfs.replication = 2`, jumlah minimum node inti adalah 2.

3. Saat Anda menggunakan Penskalaan Terkelola, Penskalaan Otomatis, atau memilih untuk mengubah ukuran klaster secara manual, sebaiknya atur `dfs.replication` ke 2 atau lebih tinggi.

Pengaturan konfigurasi	Deskripsi
dfs.replication	Jumlah simpul HDFS tempat menyalin blok tunggal (seperti blok hard drive) untuk menghasilkan lingkunga n seperti RAID. Menentukan jumlah simpul HDFS yang berisi salinan blok.
io.sort.mb	Total memori yang tersedia untuk menyortir. Nilai ini harus 10x io.sort.factor. Pengaturan ini juga dapat digunakan untuk menghitung total memori yang digunakan oleh simpul tugas dengan mencari io.sort.mb dikalikan dengan mapred.tasktracker.ap.tasks.maximum.
io.sort.spill.percent	Digunakan selama penyortiran, ketika disk akan mulai digunakan karena memori penyortiran yang dialokasikan semakin penuh.
mapred.child.java.opts	Tidak lagi digunakan. Gunakan mapred.map.child.j ava.opts dan mapred.reduce.child.java.opts sebagai gantinya. Opsi Java TaskTracker digunakan saat meluncurkan JVM untuk tugas yang akan dijalankan di dalamnya. Parameter umum adalah “-Xmx” untuk pengaturan ukuran memori maks.
mapred.map.child.java.opts	Opsi Java TaskTracker digunakan saat meluncurkan JVM untuk tugas peta yang akan dijalankan di dalamnya. Parameter umum adalah “-Xmx” untuk pengaturan ukuran timbunan memori maks.
mapred.map.tasks.speculativ e.execution	Menentukan apakah upaya tugas pemetaan dari tugas yang sama dapat diluncurkan secara paralel.

Pengaturan konfigurasi	Deskripsi
mapred.reduce.tasks.specula tive.execution	Menentukan apakah upaya tugas peredaman dari tugas yang sama dapat diluncurkan secara paralel.
mapred.map.max.attempts	Jumlah maksimum tugas pemetaan dapat dicoba. Jika semua gagal, maka tugas pemetaan ditandai sebagai gagal.
mapred.reduce.child.java.opts	Opsi Java TaskTracker digunakan saat meluncurkan JVM untuk tugas pengurangan yang akan dijalankan di dalamnya. Parameter umum adalah “-Xmx” untuk pengaturan ukuran timbunan memori maks.
mapred.reduce.max.attempts	Jumlah maksimum tugas peredaman dapat dicoba. Jika semua gagal, maka tugas pemetaan ditandai sebagai gagal.
mapred.reduce.slowstart.com pleted.maps	Jumlah tugas pemetaan yang harus diselesaikan sebelum tugas peredaman dicoba. Tidak menunggu cukup lama dapat menyebabkan kesalahan “Terlalu banyak kegagalan mengambil” dalam upaya.
mapred.reuse.jvm.num.tasks	Sebuah tugas berjalan dalam JVM tunggal. Menentukan berapa banyak tugas dapat menggunakan kembali JVM yang sama.
mapred.tasktracker.map.task s.maximum	Jumlah maksimal tugas yang dapat dieksekusi secara paralel per simpul tugas selama pemetaan.
mapred.tasktracker.reduce.t asks.maximum	Jumlah maksimal tugas yang dapat dieksekusi secara paralel per simpul tugas selama peredaman.

Jika tugas klaster Anda menggunakan banyak memori, Anda dapat meningkatkan performa dengan menggunakan lebih sedikit tugas per simpul inti dan mengurangi ukuran tumpukan pelacak pekerjaan Anda.

Langkah 7: Periksa data input

Lihatlah data input Anda. Apakah data terdistribusi secara merata di antara nilai-nilai kunci Anda? Jika data Anda sangat condong ke arah satu atau beberapa nilai kunci, beban pemrosesan dapat dipetakan ke sejumlah kecil simpul, sementara simpul lain menganggur. Distribusi pekerjaan yang tidak seimbang ini dapat mengakibatkan waktu pemrosesan yang lebih lambat.

Contoh himpunan data yang tidak seimbang adalah menjalankan klaster untuk mengurutkan kata-kata menurut abjad, tetapi memiliki himpunan data yang berisi kata-kata yang dimulai dengan huruf “a” saja. Ketika pekerjaan dipetakan, nilai pemrosesan simpul yang dimulai dengan “a” akan kewalahan, sementara simpul yang memproses kata-kata yang dimulai dengan huruf lain akan menganggur.

Memecahkan masalah klaster Lake Formation

Bagian ini memandu Anda melalui proses pemecahan masalah umum saat menggunakan Amazon EMR dengan AWS Lake Formation.

Akses danau data tidak diperbolehkan

Anda harus secara eksplisit memilih pemfilteran data pada klaster Amazon EMR sebelum Anda dapat menganalisis dan memproses data dalam danau data Anda. Ketika akses data gagal, Anda akan melihat pesan `Access is not allowed` generik dalam output entri notebook Anda.

Untuk memilih dan mengizinkan pemfilteran data di Amazon EMR, lihat Izinkan pemfilteran data di Amazon EMR di AWS Lake Formation Panduan Developer untuk melihat instruksi.

Kedaluwarsa sesi

Batas waktu sesi untuk EMR Notebooks dan Zeppelin dikendalikan oleh IAM Role untuk pengaturan `Maximum CLI/API session duration` Lake Formation. Nilai default untuk pengaturan ini adalah satu jam. Ketika sesi kehabisan waktu, Anda akan melihat pesan berikut dalam output entri notebook Anda ketika mencoba untuk menjalankan perintah Spark SQL.

```
Error 401    HTTP ERROR: 401 Problem accessing /sessions/2/statements.
Reason:  JWT token included in request failed validation.
Powered by Jetty:// 9.3.24.v20180605
 org.springframework.web.client.HttpClientErrorException: 401 JWT token included in
 request failed validation…
```

Untuk memvalidasi sesi Anda, refresh halaman. Anda akan diminta untuk mengautentikasi ulang menggunakan IdP Anda dan diarahkan kembali ke Notebook. Anda dapat terus menjalankan kueri setelah autentikasi ulang.

Tidak ada izin untuk pengguna pada tabel yang diminta

Ketika mencoba untuk mengakses tabel yang tidak dapat Anda akses, Anda akan melihat pengecualian berikut dalam output entri notebook Anda ketika mencoba untuk menjalankan perintah Spark SQL.

```
org.apache.spark.sql.AnalysisException:
 org.apache.hadoop.hive.ql.metadata.HiveException: Unable to fetch table table.
Resource does not exist or requester is not authorized to access requested
 permissions.
(Service: AWSGlue; Status Code: 400; Error Code: AccessDeniedException; Request ID: …
```

Untuk mengakses tabel, Anda harus memberikan akses ke pengguna dengan memperbarui izin yang terkait dengan tabel ini dalam Lake Formation.

Menanyakan data lintas akun yang dibagikan dengan Lake Formation

Saat Anda menggunakan Amazon EMR untuk mengakses data yang dibagikan dengan Anda dari akun lain, beberapa pustaka Spark akan mencoba memanggil operasi API. `Glue:GetUserDefinedFunctions` Karena izin AWS RAM terkelola versi 1 dan 2 tidak mendukung tindakan ini, Anda menerima pesan galat berikut:

```
"ERROR: User: arn:aws:sts::012345678901:assumed-role/my-
spark-role/i-06ab8c2b59299508a is not authorized to perform:
glue:GetUserDefinedFunctions on resource: arn:exampleCatalogResource
because no resource-based policy allows the glue:GetUserDefinedFunctions
action"
```

Untuk mengatasi kesalahan ini, administrator data lake yang membuat pembagian sumber daya harus memperbarui izin AWS RAM terkelola yang dilampirkan ke pembagian sumber daya. Versi 3 dari izin AWS RAM terkelola memungkinkan prinsipal untuk melakukan tindakan. `glue:GetUserDefinedFunctions`

Jika Anda membuat pembagian sumber daya baru, Lake Formation menerapkan versi terbaru dari izin AWS RAM terkelola secara default, dan tidak ada tindakan yang diperlukan oleh Anda.

Untuk mengaktifkan akses data lintas akun untuk pembagian sumber daya yang ada, Anda perlu memperbarui izin AWS RAM terkelola ke versi 3.

Anda dapat melihat AWS RAM izin yang ditetapkan ke sumber daya yang dibagikan dengan Anda diAWS RAM. Izin berikut disertakan dalam versi 3:

```
Databases
  AWSRAMPermissionGlueDatabaseReadWriteForCatalog
  AWSRAMPermissionGlueDatabaseReadWrite

Tables
  AWSRAMPermissionGlueTableReadWriteForCatalog
  AWSRAMPermissionGlueTableReadWriteForDatabase

AllTables
  AWSRAMPermissionGlueAllTablesReadWriteForCatalog
  AWSRAMPermissionGlueAllTablesReadWriteForDatabase
```

Untuk memperbarui versi izin AWS RAM terkelola dari pembagian sumber daya yang ada

Anda (administrator data lake) dapat memperbarui izin AWS RAM terkelola ke versi yang lebih baru dengan mengikuti petunjuk di Panduan AWS RAM Pengguna atau Anda dapat mencabut semua izin yang ada untuk jenis sumber daya dan memberikannya kembali. Jika Anda mencabut izin, AWS RAM menghapus pembagian AWS RAM sumber daya yang terkait dengan jenis sumber daya. Saat Anda memberikan kembali izin, AWS RAM buat pembagian sumber daya baru yang melampirkan versi terbaru izin terkelola. AWS RAM

Memasukkan ke dalam, membuat, dan mengubah tabel

Menyisipkan, membuat, atau mengubah tabel dalam basis data yang dilindungi oleh kebijakan Lake Formation tidak didukung. Ketika melakukan operasi ini, Anda akan melihat pengecualian berikut ini dalam output entri notebook Anda ketika mencoba untuk menjalankan perintah Spark SQL:

```
java.io.IOException:
 com.amazon.ws.emr.hadoop.fs.shaded.com.amazonaws.services.s3.model.AmazonS3Exception:
           Access Denied (Service: Amazon S3; Status Code: 403; Error Code:
 AccessDenied; Request ID: …
```

Untuk informasi selengkapnya, lihat Keterbatasan integrasi Amazon EMR dengan AWS Lake Formation.

Menulis aplikasi yang meluncurkan dan mengelola klaster

Topik

Anda dapat mengakses fungsi yang disediakan oleh Amazon EMR API dengan memanggil fungsi wrapper di salah satu AWS SDK. AWS SDK menyediakan fungsi khusus bahasa yang membungkus API layanan web dan menyederhanakan koneksi ke layanan web, menangani banyak detail koneksi untuk Anda. Untuk informasi selengkapnya tentang memanggil Amazon EMR menggunakan SDK, lihat Menggunakan SDK untuk memanggil Amazon EMR API.

> **Important**
>
> Tingkat permintaan maksimum untuk Amazon EMR adalah satu permintaan setiap sepuluh detik.

Sampel kode sumbernd-to-end Amazon EMR Java endemr Java

Developer dapat memanggil Amazon EMR API menggunakan kode Java khusus untuk melakukan hal yang sama yang mungkin dengan konsol Amazon EMR atau CLI. Bagian ini menyediakan end-to-end langkah-langkah yang diperlukan untuk menginstalAWS Toolkit for Eclipse dan menjalankan sampel kode sumber Java dengan fungsi penuh yang menambahkan langkah-langkah ke klaster Amazon EMR.

> **Note**
>
> Contoh ini berfokus pada Java, tetapi Amazon EMR juga mendukung beberapa bahasa pemrograman dengan koleksi Amazon EMR SDK. Untuk informasi selengkapnya, lihat Menggunakan SDK untuk memanggil Amazon EMR API.

Contoh kode sumber Java ini menunjukkan cara melakukan tugas-tugas berikut menggunakan Amazon EMR API:

- Mengambil kredensial AWS dan kirim ke Amazon EMR untuk melakukan panggilan API
- Mengonfigurasi langkah khusus baru dan langkah yang telah ditentukan baru
- Menambahkan langkah baru untuk klaster Amazon EMR yang ada
- Mengambil ID langkah klaster dari klaster yang berjalan

> **Note**
>
> Sampel ini menunjukkan cara menambahkan langkah-langkah untuk klaster yang ada dan dengan demikian mengharuskan Anda memiliki klaster aktif pada akun Anda.

Sebelum memulai, instal versi Eclipse IDE untuk developer Java EE yang cocok dengan platform komputer Anda. Untuk informasi lebih lanjut, kunjungi Unduhan Eclipse.

Selanjutnya, instal plugin Database Development untuk Eclipse.

Untuk menginstal plugin Database Development Eclipse

1. Buka Eclipse IDE.
2. Pilih Bantuan dan Instal Perangkat Lunak baru.
3. Dalam bidang Bekerja dengan:, ketik **`http://download.eclipse.org/releases/kepler`** atau jalur yang cocok dengan nomor versi Eclipse IDE Anda.
4. Dalam daftar item, pilih Database Development dan Selesai.
5. Mulai ulang Eclipse saat diminta.

Selanjutnya, instal Toolkit for Eclipse untuk membuat templat proyek sumber bermanfaat yang telah dikonfigurasi yang tersedia.

Untuk menginstal Toolkit for Eclipse

1. Buka Eclipse IDE.
2. Pilih Bantuan dan Instal Perangkat Lunak baru.
3. Dalam bidang Bekerja dengan:, ketik **`https://aws.amazon.com/eclipse`**.

4. Dalam daftar item, pilih AWS Toolkit for Eclipse dan Selesai.
5. Mulai ulang Eclipse saat diminta.

Selanjutnya, buat proyek AWS Java dan jalankan kode sumber sampel Java.

Untuk membuat proyek AWS Java

1. Buka Eclipse IDE.
2. Pilih File, Baru, dan Lainnya.
3. Dalam dialog Pilih wizard, pilih Proyek AWS Java dan Berikutnya.
4. Dalam dialog Proyek AWS Java Baru, di bidang **`Project name:`**, masukkan nama proyek baru Anda, misalnya **`EMR-sample-code`**.
5. Pilih Konfigurasi akun AWS..., masukkan access key publik dan privat Anda, lalu pilih Selesai. Untuk informasi lebih lanjut tentang membuat access key, lihat Bagaimana cara mendapatkan kredensial keamanan? dalam Referensi Umum Amazon Web Services.

   > **Note**
   >
   > Anda tidak boleh menyematkan access key ke dalam kode secara langsung. Amazon EMR SDK memungkinkan Anda menempatkan access key di lokasi yang diketahui sehingga Anda tidak perlu menyimpannya dalam kode.

6. Dalam proyek Java baru, klik kanan folder src, lalu pilih Baru dan Kelas.
7. Dalam dialog Kelas Java, di bidang Nama, masukkan nama untuk kelas baru Anda, misalnya **`main`**.
8. Di bagian Metode bertopik mana yang ingin Anda buat?, pilih public static void main(String [] args) dan Selesai.
9. Masukkan kode sumber Java di dalam kelas baru Anda dan tambahkan pernyataan Impor yang sesuai untuk kelas dan metode dalam sampel. Untuk kemudahan, daftar kode sumber lengkap ditampilkan di bawah ini.

   > **Note**
   >
   > Dalam kode sampel berikut, ganti contoh ID klaster (JobFlowId) *`j-xxxxxxxxxxxx`*, dengan ID klaster yang valid di akun yang Anda temukan diAWS Management Console atau dengan menggunakanAWS CLI perintah:

```
aws emr list-clusters --active | grep "Id"
```

Selain itu, ganti contoh jalur Amazon S3, *s3://path/to/my/jarfolder*, dengan jalur yang valid untuk JAR Anda. Terakhir, ganti contoh nama kelas, *com.my.Main1*, dengan nama yang benar dari kelas di JAR Anda, jika berlaku.

```
import com.amazonaws.AmazonClientException;
import com.amazonaws.auth.AWSCredentials;
import com.amazonaws.auth.AWSStaticCredentialsProvider;
import com.amazonaws.auth.profile.ProfileCredentialsProvider;
import com.amazonaws.services.elasticmapreduce.AmazonElasticMapReduce;
import com.amazonaws.services.elasticmapreduce.AmazonElasticMapReduceClientBuilder;
import com.amazonaws.services.elasticmapreduce.model.*;
import com.amazonaws.services.elasticmapreduce.util.StepFactory;

public class Main {

 public static void main(String[] args) {
  AWSCredentials credentials_profile = null;
  try {
   credentials_profile = new
 ProfileCredentialsProvider("default").getCredentials();
       } catch (Exception e) {
           throw new AmazonClientException(
                   "Cannot load credentials from .aws/credentials file. " +
                   "Make sure that the credentials file exists and the profile
 name is specified within it.",
                   e);
       }

  AmazonElasticMapReduce emr = AmazonElasticMapReduceClientBuilder.standard()
   .withCredentials(new AWSStaticCredentialsProvider(credentials_profile))
   .withRegion(Regions.US_WEST_1)
   .build();

  // Run a bash script using a predefined step in the StepFactory helper class
     StepFactory stepFactory = new StepFactory();
     StepConfig runBashScript = new StepConfig()
       .withName("Run a bash script")
```

```
       .withHadoopJarStep(stepFactory.newScriptRunnerStep("s3://jeffgoll/emr-
scripts/create_users.sh"))
       .withActionOnFailure("CONTINUE");

     // Run a custom jar file as a step
     HadoopJarStepConfig hadoopConfig1 = new HadoopJarStepConfig()
        .withJar("s3://path/to/my/jarfolder") // replace with the location of the
 jar to run as a step
        .withMainClass("com.my.Main1") // optional main class, this can be omitted
 if jar above has a manifest
        .withArgs("--verbose"); // optional list of arguments to pass to the jar
     StepConfig myCustomJarStep = new StepConfig("RunHadoopJar", hadoopConfig1);

     AddJobFlowStepsResult result = emr.addJobFlowSteps(new
 AddJobFlowStepsRequest()
    .withJobFlowId("j-xxxxxxxxxxxx") // replace with cluster id to run the steps
    .withSteps(runBashScript,myCustomJarStep));

          System.out.println(result.getStepIds());

 }
}
```

10. Pilih Jalankan, Jalankan Sebagai, dan Aplikasi Java.
11. Jika sampel berjalan dengan benar, daftar ID untuk langkah-langkah baru akan muncul di jendela konsol Eclipse IDE. Output yang benar serupa dengan berikut ini:

```
[s-39BLQZRJB2E5E, s-1L6A4ZU2SAURC]
```

Konsep umum untuk panggilan API

Topik

- Titik akhir untuk Amazon EMR
- Menentukan parameter klaster di Amazon EMR
- Availability Zone di Amazon EMR
- Cara menggunakan file tambahan dan pustaka di klaster Amazon EMR

Ketika Anda menulis aplikasi yang memanggil Amazon EMR API, ada beberapa konsep yang berlaku ketika memanggil salah satu fungsi pembungkus SDK.

Titik akhir untuk Amazon EMR

Titik akhir adalah URL yang merupakan titik masuk untuk layanan web. Setiap permintaan layanan web harus berisi titik akhir. Titik akhir menentukan Wilayah AWS tempat klaster dibuat, dijelaskan, atau dihentikan. Titik akhir ini memiliki bentuk `elasticmapreduce.`*`regionname`*`.amazonaws.com`. Jika Anda menentukan titik akhir umum (`elasticmapreduce.amazonaws.com`), Amazon EMR mengarahkan permintaan Anda ke titik akhir di Wilayah default. Untuk akun yang dibuat pada atau setelah 8 Maret 2013, Wilayah defaultnya adalah us-west-2; untuk akun lama, Wilayah defaultnya adalah us-east-1.

Untuk informasi selengkapnya tentang titik akhir untuk Amazon EMR, lihat Wilayah dan titik akhir di bagian Referensi Umum Amazon Web Services.

Menentukan parameter klaster di Amazon EMR

Parameter `Instances` memungkinkan Anda mengonfigurasi jenis dan jumlah instans EC2 untuk membuat simpul guna memproses data. Hadoop menyebarkan pemrosesan data di beberapa simpul klaster. Simpul utama bertanggung jawab untuk melacak kesehatan inti serta tugas simpul dan polling simpul untuk status hasil pekerjaan. Simpul inti dan simpul tugas melakukan pemrosesan data sebenarnya. Jika Anda memiliki klaster simpul tunggal, simpul tersebut berfungsi sebagai simpul utama dan inti.

Parameter `KeepJobAlive` dalam permintaan `RunJobFlow` menentukan apakah akan mengakhiri klaster ketika kehabisan langkah klaster untuk dieksekusi. Tetapkan nilai ini ke `False` ketika Anda tahu bahwa klaster berjalan seperti yang diharapkan. Ketika Anda memecahkan masalah alur kerja dan menambahkan langkah-langkah sementara eksekusi klaster ditangguhkan, tetapkan nilai ke `True`. Hal ini mengurangi jumlah waktu dan biaya pengunggahan hasil ke Amazon Simple Storage Service (Amazon S3), hanya untuk mengulangi proses setelah memodifikasi langkah untuk memulai ulang klaster.

Jika `KeepJobAlive` adalah `true`, setelah berhasil mendapatkan klaster untuk menyelesaikan pekerjaannya, Anda harus mengirim permintaan `TerminateJobFlows` atau klaster terus berjalan dan membuat biaya AWS.

Untuk informasi lebih lanjut tentang parameter yang unik`RunJobFlow`, lihat RunJobFlow. Untuk informasi selengkapnya tentang parameter generik dalam permintaan, lihat Parameter permintaan umum.

Availability Zone di Amazon EMR

Amazon EMR menggunakan instans EC2 sebagai simpul untuk memproses klaster. Instans EC2 memiliki lokasi yang terdiri dari Availability Zone dan Wilayah. Wilayah tersebar dan berada di wilayah geografis yang terpisah. Availability Zone adalah lokasi yang berbeda dalam Wilayah terisolasi dari kegagalan di Availability Zone lainnya. Tiap Availability Zone menyediakan konektivitas jaringan latensi rendah yang murah ke Availability Zone lain di Wilayah yang sama. Untuk daftar Wilayah dan titik akhir untuk Amazon EMR, lihat Wilayah dan titik akhir di Referensi Umum Amazon Web Services.

Parameter `AvailabilityZone` menentukan lokasi umum klaster. Parameter ini bersifat opsional dan, secara umum, kami tidak menyarankan penggunaannya. Ketika `AvailabilityZone` tidak ditentukan, Amazon EMR secara otomatis mengambil nilai `AvailabilityZone` yang terbaik untuk klaster. Anda mungkin menemukan parameter ini berguna jika Anda ingin melakukan kolokasi instans Anda dengan instans lain yang berjalan yang ada, dan klaster Anda perlu membaca atau menulis data dari instans tersebut. Untuk informasi selengkapnya, lihat Panduan Pengguna Amazon EC2 untuk Instans Linux.

Cara menggunakan file tambahan dan pustaka di klaster Amazon EMR

Ada kalanya Anda mungkin ingin menggunakan file tambahan atau pustaka khusus dengan aplikasi pemeta atau peredam Anda. Misalnya, Anda mungkin ingin menggunakan pustaka yang mengonversi file PDF menjadi teks biasa.

Untuk melakukan cache file untuk pemeta atau peredam untuk digunakan saat memakai streaming Hadoop

- Dalam bidang `args` JAR:, tambahkan argumen berikut:

  ```
  -cacheFile s3://bucket/path_to_executable#local_path
  ```

  File, `local_path`, ada di direktori kerja pemeta, yang bisa mereferensikan file.

Menggunakan SDK untuk memanggil Amazon EMR API

Topik

AWS SDK menyediakan fungsi yang membungkus API dan menangani banyak detail koneksi, seperti menghitung tanda tangan, menangani percobaan ulang permintaan, dan penanganan kesalahan. SDK juga berisi kode sampel, tutorial, dan sumber daya lain untuk membantu Anda memulai menulis aplikasi yang memanggil AWS. Memanggil fungsi wrapper dalam SDK dapat sangat menyederhanakan proses penulisan aplikasi AWS.

Untuk informasi selengkapnya tentang cara mengunduh dan menggunakan AWS SDK, lihat SDK di bawah Alat untuk Amazon Web Services.

Menggunakan AWS SDK for Java untuk membuat klaster Amazon EMR

AWS SDK for Java menyediakan tiga paket dengan fungsi Amazon EMR:

- com.amazonaws.services.elasticmapreduce
- com.amazonaws.services.elasticmapreduce.model
- com.amazonaws.services.elasticmapreduce.util

Untuk informasi selengkapnya tentang paket ini, lihat Referensi API AWS SDK for Java.

Contoh berikut menggambarkan cara SDK menyederhanakan pemrograman dengan Amazon EMR. Sampel kode di bawah ini menggunakan objek `StepFactory`, kelas pembantu untuk menciptakan jenis langkah Amazon EMR yang umum, untuk membuat klaster Hive interaktif dengan debugging diaktifkan.

```
import com.amazonaws.AmazonClientException;
import com.amazonaws.auth.AWSCredentials;
import com.amazonaws.auth.AWSStaticCredentialsProvider;
import com.amazonaws.auth.profile.ProfileCredentialsProvider;
import com.amazonaws.services.elasticmapreduce.AmazonElasticMapReduce;
import com.amazonaws.services.elasticmapreduce.AmazonElasticMapReduceClientBuilder;
import com.amazonaws.services.elasticmapreduce.model.*;
import com.amazonaws.services.elasticmapreduce.util.StepFactory;

public class Main {
```

```
public static void main(String[] args) {
 AWSCredentialsProvider profile = null;
 try {
  credentials_profile = new ProfileCredentialsProvider("default"); // specifies any
named profile in .aws/credentials as the credentials provider
       } catch (Exception e) {
           throw new AmazonClientException(
                   "Cannot load credentials from .aws/credentials file. " +
                   "Make sure that the credentials file exists and that the profile
name is defined within it.",
                   e);
       }

 // create an EMR client using the credentials and region specified in order to create
the cluster
 AmazonElasticMapReduce emr = AmazonElasticMapReduceClientBuilder.standard()
  .withCredentials(credentials_profile)
  .withRegion(Regions.US_WEST_1)
  .build();

       // create a step to enable debugging in the AWS Management Console
 StepFactory stepFactory = new StepFactory();
 StepConfig enabledebugging = new StepConfig()
     .withName("Enable debugging")
     .withActionOnFailure("TERMINATE_JOB_FLOW")
     .withHadoopJarStep(stepFactory.newEnableDebuggingStep());

       // specify applications to be installed and configured when EMR creates the
cluster
 Application hive = new Application().withName("Hive");
 Application spark = new Application().withName("Spark");
 Application ganglia = new Application().withName("Ganglia");
 Application zeppelin = new Application().withName("Zeppelin");

 // create the cluster
 RunJobFlowRequest request = new RunJobFlowRequest()
         .withName("MyClusterCreatedFromJava")
         .withReleaseLabel("emr-5.20.0") // specifies the EMR release version label,
we recommend the latest release
         .withSteps(enabledebugging)
         .withApplications(hive,spark,ganglia,zeppelin)
         .withLogUri("s3://path/to/my/emr/logs") // a URI in S3 for log files is
required when debugging is enabled
```

```
          .withServiceRole("EMR_DefaultRole") // replace the default with a custom IAM
 service role if one is used
          .withJobFlowRole("EMR_EC2_DefaultRole") // replace the default with a custom
 EMR role for the EC2 instance profile if one is used
          .withInstances(new JobFlowInstancesConfig()
              .withEc2SubnetId("subnet-12ab34c56")
              .withEc2KeyName("myEc2Key")
              .withInstanceCount(3)
              .withKeepJobFlowAliveWhenNoSteps(true)
              .withMasterInstanceType("m4.large")
              .withSlaveInstanceType("m4.large"));

     RunJobFlowResult result = emr.runJobFlow(request);
     System.out.println("The cluster ID is " + result.toString());

 }

}
```

Minimal, Anda harus melewati peran layanan dan peran alur kerja yang sesuai dengan EMR_DefaultRole dan EMR_EC2_DefaultRole, secara berurutan. Anda dapat melakukannya dengan memanggil perintah AWS CLI ini untuk akun yang sama. Pertama, periksa untuk melihat apakah peran sudah ada:

```
aws iam list-roles | grep EMR
```

Profil instans (EMR_EC2_DefaultRole) dan peran layanan (EMR_DefaultRole) akan ditampilkan jika ada:

```
"RoleName": "EMR_DefaultRole",
            "Arn": "arn:aws:iam::AccountID:role/EMR_DefaultRole"
            "RoleName": "EMR_EC2_DefaultRole",
            "Arn": "arn:aws:iam::AccountID:role/EMR_EC2_DefaultRole"
```

Jika peran default tidak ada, Anda dapat menggunakan perintah berikut untuk membuatnya:

```
aws emr create-default-roles
```

Mengelola Amazon EMR Service Quotas

Topik

Topik dalam bagian ini menjelaskan kuota layanan EMR (sebelumnya disebut sebagai batas layanan), cara mengelolanya dalamAWS Management Console, dan ketika lebih menguntungkan menggunakan CloudWatch Events, bukan kuota layanan untuk memantau klaster dan memicu tindakan.

Apa itu Amazon EMR Service Quotas

Akun AWS Anda memiliki kuota layanan default, juga dikenal sebagai batasan, untuk setiap layanan AWS. Layanan EMR memiliki dua jenis batasan:

- Batasan sumber daya - Anda dapat menggunakan EMR untuk membuat sumber daya EC2. Namun, sumber daya EC2 ini tunduk pada kuota layanan. Batasan sumber daya dalam kategori ini adalah:
  - Jumlah maksimum klaster aktif yang dapat dijalankan pada waktu yang sama.
  - Jumlah maksimum instans aktif per grup instans.
- Batasan API - Ketika menggunakan EMR API, kedua jenis batasannya adalah:
  - Batasan lonjakan – Ini adalah jumlah maksimum panggilan API yang dapat Anda lakukan sekaligus. Sebagai contoh, jumlah maksimum permintaan AddInstanceFleet API yang dapat Anda buat per detik ditetapkan pada 5 panggilan/detik sebagai default. Ini berarti bahwa batasan lonjakan AddInstanceFleet API adalah 5 panggilan/detik, atau bahwa, pada waktu tertentu, Anda dapat membuat paling banyak 5 panggilan AddInstanceFleet API. Namun, setelah Anda menggunakan batasan lonjakan, panggilan berikutnya dibatasi oleh batasan laju.
  - Batasan laju – Ini adalah laju pengisian kapasitas lonjakan API. Sebagai contoh, laju pengisian AddInstanceFleet panggilan diatur pada 0,5 panggilan/detik sebagai default. Ini berarti bahwa setelah Anda mencapai batasan lonjakan, Anda harus menunggu setidaknya 2 detik (0,5 panggilan/detik X 2 detik = 1 panggilan) untuk membuat panggilan API. Jika Anda membuat panggilan sebelum itu, Anda dibatasi oleh layanan web EMR. Pada titik mana pun, Anda hanya dapat membuat panggilan sebanyak kapasitas lonjakan tanpa dibatasi. Setiap detik

tambahan yang Anda tunggu, kapasitas lonjakan Anda meningkat sebanyak 0,5 panggilan hingga mencapai batas maksimum 5, yang merupakan batas lonjakan.

Bagaimana cara mengelola Amazon EMR Service Quotas

Service Quotas adalah fitur AWS yang dapat Anda gunakan untuk melihat dan mengelola kuota layanan Amazon EMR Anda, atau batasan, dari lokasi pusat menggunakan AWS Management Console, API atau CLI. Untuk mempelajari selengkapnya tentang melihat kuota dan meminta peningkatan, lihat AWSService Quotas dalam Referensi Umum Amazon Web.

Untuk API tertentu, mengatur CloudWatch kejadian mungkin merupakan opsi yang lebih baik daripada meningkatkan kuota layanan. Anda juga dapat menghemat waktu dengan menggunakan CloudWatch untuk mengatur alarm dan memicu peningkatan permintaan secara proaktif, sebelum mencapai kuota layanan. Untuk rincian lebih lanjut, lihat Waktu untuk mengatur kejadian EMR di CloudWatch.

Waktu untuk mengatur kejadian EMR di CloudWatch

Untuk beberapa API polling, seperti DescribeCluster, DescribeStep, dan ListClusters, mengatur CouldWatch kejadian dapat mengurangi waktu respons terhadap perubahan dan membebaskan kuota layanan Anda. Misalnya, jika Anda memiliki fungsi Lambda yang disiapkan untuk dijalankan saat status klaster berubah, seperti saat satu langkah selesai atau satu klaster berakhir, Anda dapat menggunakan pemicu tersebut untuk memulai tindakan berikutnya di alur kerja Anda, bukannya menunggu polling berikutnya. Jika tidak, jika Anda memiliki instans Amazon EC2 khusus atau fungsi Lambda yang terus-menerus melakukan polling API EMR untuk perubahan, Anda tidak hanya membuang sumber daya komputasi tetapi juga dapat mencapai kuota layanan Anda.

Berikut ini adalah beberapa kasus ketika Anda mungkin mendapatkan keuntungan dengan berpindah ke arsitektur yang didorong kejadian.

Kasus 1: Polling EMRDescribeCluster menggunakan panggilan untuk penyelesaian langkah

Example Polling EMR menggunakan panggilan DescribeCluster API untuk penyelesaian langkah

Pola umum adalah untuk mengirimkan langkah untuk klaster yang berjalan dan polling Amazon EMR untuk status tentang langkah, biasanya menggunakan DescribeCluster atau DescribeStep API. Tugas ini juga dapat dicapai dengan penundaan minimal dengan mengaitkan ke kejadian Perubahan Status Langkah Amazon EMR.

Kejadian ini mencakup informasi berikut dalam muatannya.

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

Dalam peta detail, fungsi Lambda dapat mengurai untuk "state", "stepId", atau "clusterId" untuk menemukan informasi yang relevan.

Kasus 2: Polling EMR untuk klaster yang tersedia untuk menjalankan alur kerja

Example Polling EMR untuk klaster yang tersedia untuk menjalankan alur kerja

Pola untuk pelanggan yang menjalankan beberapa klaster adalah untuk menjalankan alur kerja pada klaster segera setelah mereka tersedia. Jika terdapat banyak klaster yang berjalan dan alur kerja yang perlu dilakukan pada klaster yang menunggu, pola dapat membuat polling EMR menggunakan DescribeCluster atau panggilan ListClusters API untuk klaster yang tersedia. Cara lain untuk mengurangi keterlambatan dalam mengetahui kapan klaster siap untuk langkah, akan memproses kejadian Perubahan Status Klaster Amazon EMR.

Kejadian ini mencakup informasi berikut dalam muatannya.

```
{
  "version": "0",
```

```
  "id": "999cccaa-eaaa-0000-1111-123456789012",
  "detail-type": "EMR Cluster State Change",
  "source": "aws.emr",
  "account": "123456789012",
  "time": "2016-12-16T20:43:05Z",
  "region": "us-east-1",
  "resources": [],
  "detail": {
    "severity": "INFO",
    "stateChangeReason": "{\"code\":\"\"}",
    "name": "Development Cluster",
    "clusterId": "j-123456789ABCD",
    "state": "WAITING",
    "message": "Amazon EMR cluster j-123456789ABCD ..."
  }
}
```

Untuk kejadian ini, fungsi Lambda dapat diatur untuk segera mengirim alur kerja menunggu untuk klaster segera setelah statusnya berubah menjadi WAITING.

Kasus 3: Polling EMR untuk penghentian klaster

Example Polling EMR untuk penghentian klaster

Pola umum pelanggan yang menjalankan banyak klaster EMR adalah polling Amazon EMR untuk klaster yang diakhiri sehingga pekerjaan tidak lagi dikirim ke sana. Anda dapat menerapkan pola ini dengan DescribeCluster dan panggilan ListClusters API atau dengan menggunakan kejadian Perubahan Status Klaster Amazon EMR.

Setelah penghentian klaster, kejadian yang dipancarkan terlihat seperti contoh berikut.

```
{
  "version": "0",
  "id": "1234abb0-f87e-1234-b7b6-000000123456",
  "detail-type": "EMR Cluster State Change",
  "source": "aws.emr",
  "account": "123456789012",
  "time": "2016-12-16T21:00:23Z",
  "region": "us-east-1",
  "resources": [],
  "detail": {
    "severity": "INFO",
```

```
    "stateChangeReason": "{\"code\":\"USER_REQUEST\",\"message\":\"Terminated by user
 request\"}",
    "name": "Development Cluster",
    "clusterId": "j-123456789ABCD",
    "state": "TERMINATED",
    "message": "Amazon EMR Cluster jj-123456789ABCD (Development Cluster) has
 terminated at 2016-12-16 21:00 UTC with a reason of USER_REQUEST."
  }
}
```

Bagian "detail" muatan termasuk clusterId dan status yang dapat ditindak.

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.