aws

Panduan Developer

Amazon MQ

Amazon MQ: Panduan Developer

Table of Contents

Apa itu Amazon MQ? 1
Fitur Amazon MQ 1
Bagaimana cara memulai dengan Amazon MQ? 2
Bagaimana saya bisa memberikan umpan balik ke Amazon MQ? 3
Menyiapkan 4
Langkah 1: Prasyarat 4
Mendaftar untuk Akun AWS 4
Buat pengguna dengan akses administratif 5
Buat pengguna dan dapatkan AWS kredensialmu 6
Langkah 3: bersiap menggunakan kode contoh 8
Langkah selanjutnya 9
Memulai: Membuat dan menghubungkan ke broker ActiveMQ 10
Buat broker ActiveMQ 10
Memulai: Membuat dan menghubungkan ke broker RabbitMQ 13
Buat broker RabbitMQ 13
Mengelola broker 16
Terhubung ke Amazon MQ 16
Titik akhir layanan 16
Titik akhir broker 17
Connect ke Amazon MQ menggunakan Dual-stack (dan) endpoint IPv4 IPv6 17
Connect ke Amazon MQ menggunakan AWS PrivateLink 18
Autentikasi dan otorisasi 19
Otentikasi dan otorisasi untuk Amazon MQ untuk RabbitMQ 19
Otentikasi dan otorisasi untuk Amazon MQ untuk ActiveMQ 20
Meningkatkan versi mesin 21
Meningkatkan versi mesin secara manual 21
Memutakhirkan jenis instans 24
Penyimpanan 27
Perbedaan beragam Jenis Penyimpanan 28
Mengkonfigurasi broker pribadi 29
Mengkonfigurasi broker pribadi di Konsol Manajemen AWS 30
Mengakses konsol web broker Amazon MQ tanpa aksesibilitas publik 30
Penjadwalan pemeliharaan broker 31
Melakukan boot ulang broker 34

Untuk Melakukan Boot Ulang Broker Amazon MQ 35
Menghapus broker 35
Menghapus broker Amazon MQ 35
Status broker 36
Penandaan 37
Menambahkan tag di Amazon MQ Console 37
Amazon MQ for ActiveMQ 39
Amazon MQ untuk broker ActiveMQ 39
Pialang 39
Pengguna 42
Menyebarkan broker 43
Broker instans tunggal 43
Pialang aktif/siaga 44
Jaringan broker 45
Bagaimana cara kerja Jaringan Pialang? 45
Bagaimana Cara Jaringan Broker Menangani Kredensial? 46
Lintas wilayah 46
Failover Dinamis dengan Konektor Transportasi 48
Tipe instans 49
Konfigurasi broker 50
Atribut 51
Menggunakan file konfigurasi Spring XML 51
Membuat konfigurasi 52
Mengedit revisi konfigurasi 55
Elemen yang diizinkan 57
Atribut yang Diizinkan 60
Pengumpulan yang Diizinkan 72
Atribut Elemen Anak 79
Replikasi data Lintas Wilayah 86
Pialang primer dan replika 86
Membuat broker CRDR 87
Menghapus broker CRDR 91
Mempromosikan broker CRDR 91
Metrik 94
Tutorial ActiveMQ 96
Membuat dan mengonfigurasi jaringan broker 96

Menghubungkan aplikasi Java ke broker Anda 102
Mengintegrasikan broker ActiveMQ dengan LDAP 107
Langkah 3: (Opsional) Connect ke AWS Lambda fungsi 123
Membuat pengguna broker ActiveMQ 125
Edit pengguna broker ActiveMQ 127
Hapus pengguna broker ActiveMQ 128
Contoh kerja Java 128
Manajemen versi 140
Versi mesin yang didukung di Amazon MQ untuk ActiveMQ 140
Peningkatan versi mesin 141
Membuat daftar versi mesin yang didukung 141
Amazon MQ untuk praktik terbaik ActiveMQ 142
Jangan Pernah Memodifikasi atau Menghapus Antarmuka Jaringan Elastis Amazon MQ 142
Selalu Gunakan Pooling Koneksi 143
Selalu Gunakan Transportasi Failover untuk Terhubung ke Beberapa Titik Akhir Broker 144
Hindari Penggunaan Penyeleksi Pesan 145
Memilih Tujuan Virtual untuk Langganan Tahan Lama 145
Jika menggunakan peering VPC Amazon, hindari klien IPs dalam rentang CIDR `10.0.0.0/16` 145
Menonaktifkan Penyimpanan dan Pengiriman Bersamaan untuk Antrean dengan Konsumen Lambat 145
Memilih Tipe Instans Broker yang Tepat untuk Throughput Terbaik 146
Pilih jenis penyimpanan broker yang tepat untuk throughput terbaik 147
Mengonfigurasi Jaringan Broker dengan Benar 147
Hindari mulai ulang lambat dengan memulihkan transaksi XA yang disiapkan 148
Amazon MQ for RabbitMQ 150
Broker 150
Port listener 150
Atribut 41
Manajemen versi 151
Membuat daftar versi mesin yang didukung 152
KelinciMQ 4 153
Dukungan versi 156
Upgrade versi 156
Menyebarkan broker RabbitMQ 157
Broker instans tunggal 157

Penyebaran cluster 158
Tipe instans 160
Jenis instans untuk penerapan klaster m7g 161
Jenis instans untuk penerapan instans tunggal m7g 162
Jenis instans untuk penyebaran instance mq.m5 tunggal 163
Jenis instans untuk penerapan mq.m5 klaster 164
Pedoman ukuran 165
Batas sumber daya default 166
Batas sumber daya maksimum 169
Broker default 174
Konfigurasi broker 179
Atribut 51
Membuat konfigurasi 180
Mengedit revisi konfigurasi 183
Nilai yang dapat dikonfigurasi 184
Autentikasi dan Otorisasi 200
Otentikasi dan otorisasi sederhana 19
OAuth 2.0 otentikasi dan otorisasi 19
Otentikasi dan otorisasi IAM 19
Otentikasi dan otorisasi LDAP 19
Otentikasi dan otorisasi HTTP 20
Otentikasi sertifikat SSL 20
Otentikasi dan otorisasi sederhana 202
OAuth 2.0 otentikasi dan otorisasi 204
Otentikasi dan otorisasi IAM 206
Otentikasi dan otorisasi HTTP 207
Otentikasi sertifikat SSL 210
Otentikasi dan otorisasi LDAP 213
Plugin 215
Plugin manajemen RabbitMQ 216
Plugin shovel 216
Plugin federasi 217
Plugin pertukaran Hash yang konsisten 218
OAuth Plugin 2.0 219
Plugin LDAP 219
Plugin HTTP 219

Plugin sertifikat SSL ... 219
plugin aws ... 220
Plugin Pertukaran Topik JMS ... 220
Protokol ... 220
Dukungan JMS ... 221
Klien RabbitMQ JMS ... 221
Didukung JMS 1.1, 2.0 dan 3.1 APIs ... 221
Autentikasi dan Otorisasi ... 222
Interoperabilitas dengan antrian AMQP di RabbitMQ ... 222
Kebijakan ... 222
Antrian kuorum ... 227
Migrasi ke antrian kuorum ... 228
Konfigurasi kebijakan ... 229
Praktik terbaik ... 230
Amazon MQ untuk praktik terbaik RabbitMQ ... 230
Pengaturan broker ... 231
Keandalan pesan ... 233
Optimalisasi kinerja ... 236
Ketahanan jaringan ... 241
Tutorial RabbitMQ ... 243
Mengedit preferensi broker ... 243
Menggunakan Python Pika dengan Amazon MQ untuk RabbitMQ ... 244
Menyelesaikan sinkronisasi antrean yang dijeda ... 251
Mengurangi jumlah koneksi dan saluran ... 257
Langkah 2: Hubungkan aplikasi berbasis JVM ke broker Anda ... 258
Langkah 3: (Opsional) Connect ke AWS Lambda fungsi ... 262
Menggunakan otentikasi dan otorisasi OAuth 2.0 ... 265
Menggunakan otentikasi dan otorisasi IAM ... 273
Menggunakan otentikasi dan otorisasi LDAP ... 278
Menggunakan otentikasi dan otorisasi HTTP ... 284
Menggunakan otentikasi sertifikat SSL ... 289
Menggunakan mTL untuk AMQP dan endpoint manajemen ... 295
Menghubungkan aplikasi JMS Anda ... 301
Keamanan ... 304
Perlindungan data ... 305
Enkripsi ... 306

Enkripsi saat diam ... 306
Enkripsi saat bergerak ... 316
Identity and access management ... 317
Audiens ... 318
Mengautentikasi dengan identitas ... 318
Mengelola akses menggunakan kebijakan ... 319
Cara kerja Amazon MQ dengan IAM ... 321
Contoh kebijakan berbasis identitas ... 327
Autentikasi dan otorisasi API ... 330
Otentikasi dan otorisasi broker ... 335
AWS kebijakan terkelola ... 337
Menggunakan peran yang terhubung dengan layanan ... 339
Pemecahan Masalah ... 345
Validasi kepatuhan ... 347
Ketahanan ... 347
Keamanan infrastruktur ... 348
Praktik terbaik keamanan ... 348
Lebih memilih broker tanpa aksesibilitas publik ... 349
Selalu konfigurasikan peta otorisasi ... 349
Memblokir Protokol yang Tidak Diperlukan ... 349
Pencatatan dan pemantauan ... 351
Mengakses metrik CloudWatch ... 351
Mengakses CloudWatch metrik menggunakan Konsol Manajemen AWS ... 352
Metrik untuk ActiveMQ ... 352
Amazon MQ untuk metrik ActiveMQ ... 352
Metrik tujuan ActiveMQ (antrean dan topik) ... 358
Metrik untuk RabbitMQ ... 361
Metrik broker RabbitMQ ... 361
Dimensi untuk metrik broker RabbitMQ ... 365
Metrik simpul RabbitMQ ... 365
Dimensi untuk metrik simpul RabbitMQ ... 366
Metrik antrean RabbitMQ ... 367
Dimensi untuk metrik antrean RabbitMQ ... 367
Metrik jaringan RabbitMQ ... 368
Dimensi untuk broker RabbitMQ ... 369
Mengkonfigurasi Amazon MQ untuk log RabbitMQ ... 369

Logging panggilan API menggunakan CloudTrail 370
Informasi Amazon MQ di CloudTrail 370
Contoh Entri Berkas Log Amazon MQ 372
Mengkonfigurasi Amazon MQ untuk log ActiveMQ 374
Memahami struktur logging di CloudWatch Log 375
Tambahkan `CreateLogGroup` izin ke pengguna Amazon MQ Anda 376
Mengonfigurasi kebijakan berbasis sumber daya untuk Amazon MQ 377
Pencegahan "confused deputy" lintas layanan 378
Pemecahan masalah 380
Grup Log Tidak Muncul di CloudWatch 381
Aliran Log Tidak Muncul di Grup CloudWatch Log 381
Kuota 382
Pialang 382
Konfigurasi 383
Pengguna 384
Penyimpanan Data 385
Throttling API 386
Pemecahan masalah 387
Memecahkan masalah ActiveMQ di Amazon MQ 387
Memecahkan masalah RabbitMQ di Amazon MQ 387
Pemecahan Masalah: Amazon MQ Umum 390
Saya tidak dapat terhubung ke konsol web broker atau titik akhir saya. 390
Pengecualian SSL 396
Saya membuat broker tetapi pembuatan broker gagal. 396
Broker saya memulai kembali dan saya tidak yakin mengapa. 396
Memecahkan masalah ActiveMQ di Amazon MQ 397
Mengambil CloudWatch Log 398
Menghubungkan ke broker setelah restart 398
Beberapa klien tidak dapat terhubung 399
`JSP`pengecualian di konsol web 400
Pemecahan masalah: RabbitMQ di Amazon MQ 400
Saya tidak dapat melihat metrik untuk antrian atau host virtual saya di. CloudWatch 401
Bagaimana cara mengaktifkan plugin di RabbitMQ di Amazon MQ? 401
Saya tidak dapat mengubah konfigurasi VPC Amazon untuk broker. 401
Penerapan cluster telah menghentikan sinkronisasi antrian saya. 401
Amazon MQ saya untuk broker instans tunggal RabbitMQ sedang dalam loop restart. 402

Saya kehilangan akses ke semua akun administrator di broker saya. 402
BROKER_ENI_DELETED 402
BROKER_OOM 403
RABBITMQ_MEMORY_ALARM 404
Langkah 1: Diagnosis alarm memori tinggi 405
Langkah 2: Alamat dan cegah alarm memori tinggi 407
RABBITMQ_INVALID_KMS_KEY 409
Mendiagnosis dan menangani INVALID_KMS_KEY 409
RABBITMQ_DISK_ALARM 410
Mendiagnosis dan menangani alarm batas disk 410
RABBITMQ_CLUSTER_DISK_USAGE_TOO_HIGH_FOR_INSTANCE_CHANGE 411
Mendiagnosis dan menangani alarm perubahan tipe instance 411
RABBITMQ_INVALID_ASSUMEROLE 412
Mendiagnosis dan menangani RABBITMQ_INVALID_ASSUMEROLE 412
RABBITMQ_INVALID_ARN_LDAP 413
Mendiagnosis dan menangani RABBITMQ_INVALID_ARN_LDAP 414
RABBITMQ_INVALID_ARN_HTTP 415
Mendiagnosis dan menangani RABBITMQ_INVALID_ARN_HTTP 415
RABBITMQ_INVALID_ARN_SSL 416
Mendiagnosis dan menangani RABBITMQ_INVALID_ARN_SSL 416
RABBITMQ_INVALID_ARN 417
Mendiagnosis dan menangani RABBITMQ_INVALID_ARN 418
Sumber daya terkait 419
Sumber daya Amazon MQ 419
Sumber daya Amazon MQ for ActiveMQ 420
Sumber daya Amazon MQ for RabbitMQ 420
Catatan rilis 422
..... cdlx

Apa itu Amazon MQ?

Amazon MQ adalah layanan broker pesan terkelola untuk Apache ActiveMQ Classic dan RabbitMQ yang mengelola pengaturan, pengoperasian, dan pemeliharaan broker pesan. Anda dapat membuat broker Amazon MQ baru menggunakan protokol pesan standar industri, atau memigrasikan pialang pesan yang ada ke Amazon MQ tanpa menulis ulang kode pesan.

Broker adalah lingkungan broker pesan yang berjalan di Amazon MQ. Ini adalah blok bangunan dasar Amazon MQ. Broker pesan memungkinkan aplikasi dan komponen perangkat lunak untuk berkomunikasi menggunakan berbagai bahasa pemrograman, sistem operasi, dan protokol olahpesan formal. Anda dapat menggunakan broker Amazon MQ untuk komunikasi antara skala besar, aplikasi dan komponen cloud native.

Topik

- Fitur Amazon MQ
- Bagaimana cara memulai dengan Amazon MQ?
- Bagaimana saya bisa memberikan umpan balik ke Amazon MQ?

Fitur Amazon MQ

Pemeliharaan terkelola dan peningkatan versi

Amazon MQ melakukan pemeliharaan dan peningkatan versi untuk broker pesan selama jendela pemeliharaan terjadwal Anda.

Pantau broker dengan CloudWatch

Amazon MQ terintegrasi dengan Amazon CloudWatch sehingga Anda dapat melihat dan menganalisis metrik untuk broker dan antrian Anda. Anda dapat melihat dan menganalisis metrik dari konsol Amazon MQ, konsol, CloudWatch baris perintah, dan API. Metrik secara otomatis dikumpulkan dan didorong ke CloudWatch setiap menit.

Keamanan

Amazon MQ menyediakan enkripsi pesan Anda saat istirahat dan dalam perjalanan. Koneksi ke broker menggunakan SSL, dan akses dapat dibatasi ke titik akhir pribadi dalam VPC Amazon Anda.

Selain itu, Anda dapat menggunakan AWS Identity and Access Management(IAM) untuk mengontrol tindakan yang dapat dilakukan pengguna dan grup IAM Anda pada broker MQ Amazon tertentu.

Antrian kuorum untuk RabbitMQ di Amazon MQ

Quorum queues adalah tipe antrian yang direplikasi yang terdiri dari node pemimpin (replika primer) dan node pengikut (replika lainnya). Setiap node berada di zona ketersediaan yang berbeda, jadi jika satu node sementara tidak tersedia, pengiriman pesan berlanjut dengan replika pemimpin yang baru dipilih di zona ketersediaan lain. Antrian kuorum berguna untuk menangani pesan racun, yang terjadi ketika pesan gagal dan diminta ulang beberapa kali.

Replikasi data lintas wilayah untuk ActiveMQ di Amazon MQ

Replikasi data Cross-Region (CRDR) memungkinkan replikasi pesan asinkron dari broker utama di Wilayah utama ke broker replika di AWS Wilayah replika. Dengan mengeluarkan permintaan failover ke Amazon MQ API, broker replika saat ini dipromosikan ke peran broker utama, dan broker utama saat ini diturunkan ke peran replika.

Bagaimana cara memulai dengan Amazon MQ?

Untuk memulai ActiveMQ di Amazon MQ, tinjau dokumentasi berikut:

- Memulai: Membuat dan menghubungkan ke broker ActiveMQ
- the section called “Menyebarkan broker”
- Tutorial ActiveMQ
- the section called “Amazon MQ untuk praktik terbaik ActiveMQ”

Untuk memulai RabbitMQ di Amazon MQ, tinjau dokumentasi berikut:

- Memulai: Membuat dan menghubungkan ke broker RabbitMQ
- the section called “Menyebarkan broker RabbitMQ”
- the section called “Tutorial RabbitMQ”
- the section called “Amazon MQ untuk praktik terbaik RabbitMQ”

Untuk mempelajari tentang Amazon MQ REST APIs, lihat Referensi API Amazon MQ REST.

Untuk mempelajari tentang AWS CLI perintah Amazon MQ, lihat Amazon MQ di AWS CLI Referensi Perintah.

Bagaimana saya bisa memberikan umpan balik ke Amazon MQ?

Kami menyambut dan mendorong umpan balik Anda tentang dokumentasi. Anda dapat menggunakan ikon jempol ke atas dan jempol ke bawah di sisi kanan untuk mengirimkan umpan balik, atau Anda dapat menggunakan formulir “Berikan umpan balik” yang ditautkan di bawah ini.

Untuk menghubungi tim Amazon MQ, gunakan Forum Diskusi Amazon MQ.

Menyiapkan Amazon MQ

Sebelum Anda dapat menggunakan Amazon MQ, Anda harus menyelesaikan langkah-langkah berikut.

Topik

- Langkah 1: Prasyarat
- Langkah 2: buat pengguna dan dapatkan AWS kredensialnya
- Langkah 3: bersiap menggunakan kode contoh
- Langkah selanjutnya

Langkah 1: Prasyarat

Mendaftar untuk Akun AWS

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

Untuk mendaftar untuk Akun AWS

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

   Bagian dari prosedur pendaftaran melibatkan menerima panggilan telepon atau pesan teks dan memasukkan kode verifikasi pada keypad telepon.

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

AWS mengirimi Anda email konfirmasi setelah proses pendaftaran selesai. Kapan saja, Anda dapat melihat aktivitas akun Anda saat ini dan mengelola akun Anda dengan masuk https://aws.amazon.comke/ dan memilih Akun Saya.

Buat pengguna dengan akses administratif

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

Amankan Anda Pengguna root akun AWS

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

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

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

Buat pengguna dengan akses administratif

1. Aktifkan Pusat Identitas IAM.

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

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

Masuk sebagai pengguna dengan akses administratif

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

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

Tetapkan akses ke pengguna tambahan

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

   Untuk petunjuknya, lihat Membuat set izin di Panduan AWS IAM Identity Center Pengguna.
2. Tetapkan pengguna ke grup, lalu tetapkan akses masuk tunggal ke grup.

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

Langkah 2: buat pengguna dan dapatkan AWS kredensialnya

Pengguna membutuhkan akses terprogram jika mereka ingin berinteraksi dengan AWS luar. Konsol Manajemen AWS Cara untuk memberikan akses terprogram tergantung pada jenis pengguna yang mengakses AWS.

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

Pengguna mana yang membutuhkan akses programatis?	Untuk	Oleh
IAM	(Disarankan) Gunakan kredenal konsol sebagai kredensil sementara untuk menandatangani permintaan terprogram ke,, atau. AWS CLI AWS SDKs AWS APIs	Mengikuti petunjuk untuk antarmuka yang ingin Anda gunakan. • Untuk itu AWS CLI, lihat Login untuk pengembangan AWS lokal di Panduan AWS Command Line Interface Pengguna. • Untuk AWS SDKs, lihat Login untuk pengembangan AWS lokal di Panduan Referensi Alat AWS SDKs dan Alat.

Pengguna mana yang membutuhkan akses programatis?	Untuk	Oleh
Identitas tenaga kerja (Pengguna yang dikelola di Pusat Identitas IAM)	Gunakan kredensi sementara untuk menandatangani permintaan terprogram ke AWS CLI,, AWS SDKs atau. AWS APIs	Mengikuti petunjuk untuk antarmuka yang ingin Anda gunakan. • Untuk AWS CLI, lihat Mengkonfigurasi yang akan AWS CLI digunakan AWS IAM Identity Center dalam Panduan AWS Command Line Interface Pengguna. • Untuk AWS SDKs, alat, dan AWS APIs, lihat Autentikasi Pusat Identitas IAM di Panduan Referensi Alat AWS SDKs dan Alat.
IAM	Gunakan kredensi sementara untuk menandatangani permintaan terprogram ke AWS CLI,, AWS SDKs atau. AWS APIs	Mengikuti petunjuk dalam Menggunakan kredensil sementara dengan AWS sumber daya di Panduan Pengguna IAM.

Pengguna mana yang membutuhkan akses programatis?	Untuk	Oleh
IAM	(Tidak direkomendasikan) Gunakan kredensi jangka panjang untuk menandatangani permintaan terprogram ke AWS CLI,, AWS SDKs atau. AWS APIs	Mengikuti petunjuk untuk antarmuka yang ingin Anda gunakan. • Untuk mengetahui AWS CLI, lihat Mengautentikasi menggunakan kredensil pengguna IAM di Panduan Pengguna.AWS Command Line Interface • Untuk AWS SDKs dan alat, lihat Mengautentikasi menggunakan kredensil jangka panjang di Panduan Referensi Alat AWS SDKs dan Alat. • Untuk AWS APIs, lihat Mengelola kunci akses untuk pengguna IAM di Panduan Pengguna IAM.

Langkah 3: bersiap menggunakan kode contoh

Tutorial berikut menunjukkan bagaimana Anda dapat bekerja dengan broker Amazon MQ menggunakan Konsol Manajemen AWS serta cara terhubung ke Amazon MQ Anda untuk ActiveMQ dan Amazon MQ untuk broker RabbitMQ secara terprogram. Untuk menggunakan kode contoh ActiveMQ Java, Anda harus menginstal Alat Pengembangan Java Standard Edition dan membuat beberapa perubahan pada kode.

Anda juga dapat membuat dan mengelola broker secara terprogram menggunakan Amazon MQ REST API dan. AWS SDKs

Langkah selanjutnya

Setelah Anda siap bekerja dengan Amazon MQ, mulailah dengan Membuat broker. Tergantung pada jenis mesin broker, Anda dapat menghubungkan aplikasi Java ke broker Amazon MQ for ActiveMQ atau menggunakan pustaka klien RabbitMQ Java untuk menghubungkan aplikasi berbasis JVM ke broker Amazon MQ for RabbitMQ.

Memulai: Membuat dan menghubungkan ke broker ActiveMQ

Broker adalah lingkungan broker pesan yang berjalan di Amazon MQ. Ini adalah blok bangunan dasar Amazon MQ. Deskripsi gabungan dari kelas instance broker (`m5`) dan size (`large`,`medium`) disebut tipe instance broker (misalnya,`mq.m5.large`). Untuk informasi selengkapnya, lihat Apa itu Amazon MQ untuk broker ActiveMQ?.

Buat broker ActiveMQ

Tugas Amazon MQ yang pertama dan paling umum adalah membuat broker. Contoh berikut menunjukkan bagaimana Anda dapat menggunakan Konsol Manajemen AWS untuk membuat broker dasar.

1. Masuk ke konsol Amazon MQ.
2. Di halaman Pilih mesin broker, pilih Apache ActiveMQ.
3. Di halaman Pilih deployment dan penyimpanan, pada bagian Mode deployment dan jenis penyimpanan, lakukan hal berikut:

   a. Pilih Mode deployment (misalnya, Broker aktif/siaga). Untuk informasi selengkapnya, lihat Opsi penyebaran untuk Amazon MQ untuk broker ActiveMQ.

      - Broker Single-instance terdiri dari satu broker dalam satu Availability Zone. Broker berkomunikasi dengan aplikasi Anda dan dengan volume penyimpanan Amazon EBS atau Amazon EFS. Untuk informasi selengkapnya, lihat Opsi 1: Pialang instans tunggal Amazon MQ.
      - Broker aktif/siaga untuk ketersediaan tinggi terdiri dari dua broker di dua Availability Zone yang berbeda, dikonfigurasi dalam pasangan redundan. Broker ini berkomunikasi secara sinkron dengan aplikasi Anda dan Amazon EFS. Untuk informasi selengkapnya, lihat Opsi 2: active/standby Broker Amazon MQ untuk ketersediaan tinggi.

   b. Pilih Jenis penyimpanan (misalnya, EBS). Untuk informasi selengkapnya, lihat Storage.

> **Note**
>
> Amazon EBS mereplikasi data dalam satu Availability Zone dan tidak mendukung mode deployment ActiveMQ aktif/siaga.

c. Pilih Selanjutnya.

4. Di halaman Konfigurasi pengaturan, pada bagian Detail, lakukan hal berikut:

   a. Masukkan nama Broker.

> **Important**
>
> Jangan menambahkan informasi identitas pribadi (PII) atau informasi rahasia atau sensitif lainnya dalam nama broker. Nama broker dapat diakses oleh AWS layanan lain, termasuk CloudWatch Log. Nama broker tidak dimaksudkan untuk digunakan untuk data pribadi atau sensitif.

> **Note**
>
> Di bagian Pengaturan tambahan, Anda juga dapat mengonfigurasi yang berikut:
>
> - Konfigurasi
> - CloudWatch log
> - Akses pribadi
> - Jendela pemeliharaan broker

   b. Pilih Tipe instans broker (misalnya, mq.m5.large). Untuk informasi selengkapnya, lihat Broker instance types.

5. Di bagian Akses Konsol Web ActiveMQ, sediakan Nama pengguna dan Kata sandi. Pembatasan berikut berlaku untuk nama pengguna dan kata sandi broker:

   - Nama pengguna Anda hanya dapat berisi karakter alfanumerik, tanda hubung, titik, garis bawah, dan tilde (- . _ ~).
   - Kata sandi Anda setidaknya harus terdiri dari 12 karakter, berisi setidaknya 4 karakter unik, dan tidak boleh berisi koma, titik dua, atau tanda yang sama (,:=).

> **Important**
>
> Jangan menambahkan informasi identitas pribadi (PII) atau informasi rahasia atau sensitif lainnya dalam nama pengguna broker. Nama pengguna broker dapat diakses oleh AWS layanan lain, termasuk CloudWatch Log. Nama pengguna broker tidak dimaksudkan untuk digunakan untuk data pribadi atau sensitif.

6. Pilih Men-deploy.

   Status Pembuatan sedang berlangsung ditampilkan saat Amazon MQ membuat broker Anda.

   Pembuatan broker berlangsung sekitar 15 menit.

   Saat broker berhasil dibuat, Amazon MQ menampilkan status Berjalan.

7. Pilih ***MyBroker***.

   Pada ***MyBroker***halaman, di bagian Connect, perhatikan URL konsol web ActiveMQ broker Anda, misalnya:

   ```
   https://b-1234a5b6-78cd-901e-2fgh-3i45j6k178l9-1.mq.us-east-2.amazonaws.com:8162
   ```

   Perhatikan juga Titik Akhir protokol tingkat wire. Berikut ini adalah contoh dari OpenWire endpoint:

   ```
   ssl://b-1234a5b6-78cd-901e-2fgh-3i45j6k178l9-1.mq.us-east-2.amazonaws.com:61617
   ```

Memulai: Membuat dan menghubungkan ke broker RabbitMQ

Broker adalah lingkungan broker pesan yang berjalan di Amazon MQ. Ini adalah blok bangunan dasar Amazon MQ. Deskripsi gabungan dari kelas instance broker (`m5`) dan size (`large`,`medium`) disebut tipe instance broker (misalnya,`mq.m5.large`). Untuk informasi selengkapnya, lihat Apa itu Amazon MQ untuk broker RabbitMQ?

Buat broker RabbitMQ

Tugas Amazon MQ yang pertama dan paling umum adalah membuat broker. Contoh berikut menunjukkan bagaimana Anda dapat menggunakan Konsol Manajemen AWS untuk membuat broker dasar.

Saat membuat Amazon MQ untuk broker RabbitMQ, ikuti praktik terbaik penyiapan broker untuk RabbitMQ untuk memaksimalkan kinerja broker dan mengoptimalkan efisiensi throughput pesan.

1. Masuk ke konsol Amazon MQ.
2. Di halaman Pilih mesin broker, pilih RabbitMQ, lalu pilih Selanjutnya.
3. Di halaman Pilih mode deployment, pilih Mode deployment, misalnya, Deployment klaster, lalu pilih Selanjutnya.

   - Broker instans tunggal terdiri dari satu broker di satu Availability Zone di balik Penyeimbang Beban Jaringan (NLB). Broker berkomunikasi dengan aplikasi Anda dan dengan volume penyimpanan Amazon EBS. Untuk informasi selengkapnya, lihat Opsi 1: Amazon MQ untuk broker instans tunggal RabbitMQ.
   - Deployment klaster RabbitMQ untuk ketersediaan tinggi adalah pengelompokan logis dari tiga node broker RabbitMQ di balik Penyeimbang Beban Jaringan, masing-masing membagikan pengguna, antrean, dan status terdistribusi di beberapa Availability Zone (AZ). Untuk informasi selengkapnya, lihat Opsi 2: Amazon MQ untuk penyebaran cluster RabbitMQ.
4. Di halaman Konfigurasi pengaturan, pada bagian Detail, lakukan hal berikut:

   a. Masukkan nama Broker.

> **Important**
>
> Jangan menambahkan informasi identitas pribadi (PII) atau informasi rahasia atau sensitif lainnya dalam nama broker. Nama broker dapat diakses oleh AWS layanan lain, termasuk CloudWatch Log. Nama broker tidak dimaksudkan untuk digunakan untuk data pribadi atau sensitif.

b. Pilih jenis instance Broker (misalnya, mq.m7g.large). Untuk informasi selengkapnya, lihat Broker instance types.

5. Di halaman Konfigurasi pengaturan, pada bagian Akses RabbitMQ, berikan Nama pengguna dan Kata sandi. Pembatasan berikut berlaku untuk kredensi masuk broker:

   - Nama pengguna Anda hanya dapat berisi karakter alfanumerik, tanda hubung, titik, dan garis bawah (- . _). Nilai ini tidak boleh berisi karakter tilde (~). Amazon MQ melarang penggunaan `guest` sebagai nama pengguna.
   - Kata sandi Anda setidaknya harus terdiri dari 12 karakter, berisi setidaknya 4 karakter unik, dan tidak boleh berisi koma, titik dua, atau tanda yang sama (,:=).

   > **Important**
   >
   > Jangan menambahkan informasi identitas pribadi (PII) atau informasi rahasia atau sensitif lainnya dalam nama pengguna broker. Nama pengguna broker dapat diakses oleh AWS layanan lain, termasuk CloudWatch Log. Nama pengguna broker tidak dimaksudkan untuk digunakan untuk data pribadi atau sensitif.

   > **Note**
   >
   > Di bagian Pengaturan tambahan, Anda juga dapat mengonfigurasi yang berikut:
   >
   > - Konfigurasi
   > - CloudWatch log
   > - Akses pribadi
   > - Jendela pemeliharaan broker

6. Pilih Berikutnya.
7. Di halaman Tinjau dan buat, Anda dapat meninjau pilihan dan mengeditnya sesuai kebutuhan.
8. Pilih Buat broker.

   Status Pembuatan sedang berlangsung ditampilkan saat Amazon MQ membuat broker Anda.

   Pembuatan broker berlangsung sekitar 15 menit.

   Saat broker berhasil dibuat, Amazon MQ menampilkan status Berjalan.
9. Pilih ***MyBroker***.

   Pada ***MyBroker***halaman, di bagian Connect, catat URL konsol web RabbitMQ broker Anda, misalnya:

   ```
   https://b-c8349341-ec91-4a78-ad9c-a57f23f235bb.mq.us-west-2.on.aws
   ```

   Perhatikan juga Titik Akhir secure-AMQP. Berikut adalah contoh titik akhir `amqps` yang mengekspos port listener `5671`.

   ```
   amqps://b-c8349341-ec91-4a78-ad9c-a57f23f235bb.mq.us-west-2.on.aws:5671
   ```

Mengelola broker Amazon MQ

Setelah Anda membuat broker, Anda dapat mengelola dan memelihara berbagai komponen broker Amazon MQ Anda.

Topik

- Terhubung ke Amazon MQ
- Otentikasi dan otorisasi untuk broker Amazon MQ
- Meningkatkan versi mesin broker Amazon MQ
- Memutakhirkan jenis instans broker Amazon MQ
- Amazon MQ untuk jenis penyimpanan ActiveMQ
- Mengkonfigurasi broker MQ Amazon pribadi
- Menjadwalkan jendela pemeliharaan untuk broker Amazon MQ
- Melakukan boot ulang broker Amazon MQ
- Menghapus broker Amazon MQ
- Status broker Amazon MQ
- Menambahkan tag ke sumber daya Amazon MQ

Terhubung ke Amazon MQ

Anda dapat terhubung ke Amazon MQ dari AWS layanan lain menggunakan titik akhir layanan dan titik akhir broker.

Titik akhir layanan

Metode koneksi berikut digunakan untuk API layanan Amazon MQ:

Domain	Metode koneksi
`mq.region.amazonaws.com`	IPv4
`mq.region.api.aws`	Tumpukan ganda (dan) IPv4 IPv6
`mq-fips.region.amazonaws.com`	FIPS dengan hanya IPv4

Domain	Metode koneksi
`mq-fips.region.api.aws`	FIPS dengan Dual-stack

Titik akhir broker

Metode koneksi berikut digunakan untuk broker Amazon MQ:

Domain	Metode koneksi
`brokerId.mq.region.amazonaws.com`	IPv4
`brokerId.mq.region.on.aws`	Tumpukan ganda (dan) IPv4 IPv6 **Note** Amazon MQ untuk broker ActiveMQ tidak mendukung dual-stack.

Connect ke Amazon MQ menggunakan Dual-stack (dan) endpoint IPv4 IPv6

Titik akhir dual-stack mendukung keduanya IPv4 dan lalu lintas. IPv6 Saat Anda membuat permintaan ke titik akhir dual-stack, URL endpoint akan diselesaikan ke alamat atau alamat. IPv4 IPv6 Untuk informasi selengkapnya tentang dual-stack dan titik akhir FIPS, lihat panduan Referensi SDK.

Amazon MQ mendukung titik akhir tumpukan ganda Regional, yang berarti Anda harus menentukan AWS Wilayah sebagai bagian dari nama titik akhir. Nama titik akhir tumpukan ganda menggunakan konvensi penamaan berikut:. `mq.region.api.aws` Misalnya, nama titik akhir tumpukan ganda untuk Wilayah `eu-west-1` adalah `mq.eu-west-1.api.aws`.

Untuk daftar lengkap titik akhir Amazon MQ, lihat Referensi Umum.AWS

Connect ke Amazon MQ menggunakan AWS PrivateLink

AWS PrivateLinkendpoint untuk Amazon MQ API dengan dukungan IPv4 untuk IPv6 dan menyediakan konektivitas pribadi antara virtual private cloud VPCs () dan Amazon MQ API tanpa mengekspos lalu lintas Anda ke internet publik.

> **Note**
>
> Support for hanya PrivateLink tersedia untuk titik akhir Amazon MQ API, bukan titik akhir broker. Untuk informasi lebih lanjut tentang menghubungkan secara pribadi ke titik akhir broker, lihat. Configuring a private Amazon MQ broker

Untuk mengakses Amazon MQ API menggunakan PrivateLink, Anda harus terlebih dahulu membuat titik akhir VPC antarmuka di VPC tertentu yang ingin Anda sambungkan. Saat Anda membuat titik akhir VPC, gunakan nama layanan `com.amazonaws.`*`region`*`.mq` atau `com.amazonaws.`*`region`*`.mq-fips` untuk titik akhir FIPS.

Saat Anda memanggil Amazon MQ menggunakan AWS CLI atau SDK, Anda harus menentukan URL titik akhir untuk menggunakan nama domain dual-stack: atau. `mq.`*`region`*`.api.aws` `mq-fips.`*`region`*`.api.aws` PrivateLink untuk Amazon MQ tidak mendukung nama domain default yang diakhiri dengan. `amazonaws.com` Untuk informasi selengkapnya, lihat Dual-stack dan titik akhir FIPS di Panduan Referensi SDK.

Contoh CLI berikut menunjukkan cara memanggil Wilayah Asia Pasifik (Sydney) melalui titik akhir VPC Amazon MQ. `describe-broker-engine-type`

```
AWS_USE_DUALSTACK=true aws mq describe-broker-engine-types --region ap-southeast-2
```

Untuk cara lain untuk mengonfigurasi titik akhir di CLI, lihat Menggunakan titik akhir di CLI AWS

Anda juga dapat menentukan akses pengguna ke titik akhir VPC menggunakan kebijakan titik akhir VPC. Untuk informasi selengkapnya, lihat Mengontrol akses ke titik akhir VPC menggunakan kebijakan titik akhir.

Otentikasi dan otorisasi untuk broker Amazon MQ

Amazon MQ menawarkan beberapa metode otentikasi dan otorisasi untuk mengamankan infrastruktur pesan Anda sesuai dengan kebutuhan organisasi Anda.

Otentikasi dan otorisasi untuk Amazon MQ untuk RabbitMQ

Amazon MQ untuk RabbitMQ mendukung metode otentikasi dan otorisasi berikut:

Otentikasi dan otorisasi sederhana

Dalam metode ini, pengguna broker disimpan secara internal di broker RabbitMQ dan dikelola melalui konsol web atau API manajemen. Izin untuk vhost, pertukaran, antrian, dan topik dikonfigurasi langsung di RabbitMQ. Ini adalah metode default. Untuk informasi selengkapnya, lihat Otentikasi dan otorisasi sederhana.

OAuth 2.0 otentikasi dan otorisasi

Dalam metode ini, pengguna broker dan izin mereka dikelola oleh penyedia identitas OAuth 2.0 eksternal (iDP). Otentikasi pengguna dan izin sumber daya untuk vhost, pertukaran, antrian, dan topik dipusatkan melalui sistem lingkup penyedia 2.0. OAuth Ini menyederhanakan manajemen pengguna dan memungkinkan integrasi dengan sistem identitas yang ada. Untuk informasi selengkapnya, lihat otentikasi dan otorisasi OAuth 2.0.

Otentikasi dan otorisasi IAM

Dalam metode ini, pengguna broker mengautentikasi menggunakan kredensyal AWS IAM melalui federasi keluar IAM. Kredensyal IAM digunakan untuk mendapatkan token JWT dari AWS Security Token Service (STS), dan token JWT ini berfungsi sebagai token 2.0 untuk otentikasi. OAuth Metode ini memanfaatkan dukungan OAuth 2.0 yang ada di Amazon MQ untuk RabbitMQ, AWS di mana bertindak sebagai penyedia identitas 2.0. OAuth Otentikasi pengguna ditangani oleh AWS IAM, sementara izin sumber daya untuk vhost, pertukaran, antrian, dan topik dikelola melalui kebijakan IAM dan alias ruang lingkup yang dikonfigurasi di RabbitMQ. Untuk informasi selengkapnya, lihat autentikasi dan otorisasi IAM.

Otentikasi dan otorisasi LDAP

Dalam metode ini, pengguna broker dan izin mereka dikelola oleh layanan direktori LDAP eksternal. Otentikasi pengguna dan izin sumber daya dipusatkan melalui server LDAP, memungkinkan

pengguna untuk mengakses RabbitMQ menggunakan kredensyal layanan direktori yang ada. Untuk informasi selengkapnya, lihat otentikasi dan otorisasi LDAP.

Otentikasi dan otorisasi HTTP

Dalam metode ini, pengguna broker dan izin mereka dikelola oleh server HTTP eksternal. Otentikasi pengguna dan izin sumber daya dipusatkan melalui server HTTP, memungkinkan pengguna untuk mengakses RabbitMQ menggunakan penyedia Otentikasi dan Otorisasi mereka sendiri. Untuk informasi selengkapnya tentang metode ini, lihat otentikasi dan otorisasi HTTP.

Otentikasi sertifikat SSL

Amazon MQ mendukung TLS bersama (MTLS) untuk broker RabbitMQ. Plugin otentikasi SSL menggunakan sertifikat klien dari koneksi mTLS untuk mengautentikasi pengguna. Dalam metode ini, pengguna broker diautentikasi menggunakan sertifikat klien X.509 alih-alih kredensyal nama pengguna dan kata sandi. Sertifikat klien divalidasi terhadap Otoritas Sertifikat (CA) tepercaya, dan nama pengguna diekstraksi dari bidang dalam sertifikat, seperti Nama Umum (CN) atau Nama Alternatif Subjek (SAN). Metode ini memberikan otentikasi yang kuat tanpa mentransmisikan kredensyal melalui jaringan. Untuk informasi selengkapnya, lihat otentikasi sertifikat SSL.

> **Note**
>
> RabbitMQ mendukung beberapa metode otentikasi dan otorisasi untuk digunakan secara bersamaan. Misalnya, Anda dapat mengaktifkan otentikasi OAuth 2.0 dan sederhana (internal). Untuk informasi lebih lanjut, lihat bagian tutorial OAuth 2.0 tentang mengaktifkan otentikasi OAuth 2.0 dan sederhana (internal) dan dokumentasi kontrol akses RabbitMQ. Amazon MQ merekomendasikan untuk membuat pengguna internal saat menguji konfigurasi otentikasi. Hal ini memungkinkan konfigurasi akses untuk divalidasi menggunakan RabbitMQ management API. Untuk informasi selengkapnya, lihat Validasi akses.

Otentikasi dan otorisasi untuk Amazon MQ untuk ActiveMQ

Amazon MQ untuk ActiveMQ mendukung metode otentikasi dan otorisasi berikut:

Otentikasi dan otorisasi sederhana

Dalam metode ini, pengguna broker dibuat dan dikelola melalui konsol Amazon MQ atau API. Pengguna dapat dikonfigurasi dengan izin khusus untuk mengakses antrian, topik, dan ActiveMQ

Web Console. Untuk informasi selengkapnya tentang metode ini, lihat Membuat pengguna broker ActiveMQ.

Otentikasi dan otorisasi LDAP

Dalam metode ini, pengguna broker mengautentikasi melalui kredensyal yang disimpan di server LDAP Anda. Anda dapat menambahkan, menghapus, dan memodifikasi pengguna dan menetapkan izin untuk topik dan antrian melalui server LDAP, menyediakan otentikasi dan otorisasi terpusat. Untuk informasi lebih lanjut tentang metode ini, lihat Mengintegrasikan broker ActiveMQ dengan LDAP.

Meningkatkan versi mesin broker Amazon MQ

Amazon MQ secara teratur menyediakan versi mesin broker baru untuk semua jenis mesin broker yang didukung. Versi mesin baru termasuk patch keamanan, perbaikan bug, dan peningkatan mesin broker lainnya.

Amazon MQ mengatur nomor versi sesuai dengan spesifikasi versi semantik sebagai. `X.Y.Z` Dalam implementasi Amazon MQ, `X` menunjukkan versi utama, `Y` mewakili versi minor, dan `Z` menunjukkan nomor versi patch. Amazon MQ mendukung dua jenis peningkatan:

- Upgrade versi utama - Terjadi ketika nomor versi mesin utama berubah. Misalnya, memutakhirkan dari RabbitMQ versi 3 .13 ke versi 4 .2 dianggap sebagai peningkatan versi utama.
- Peningkatan versi minor - Terjadi ketika hanya nomor versi mesin minor yang berubah. Misalnya, memutakhirkan dari versi 3. 11 ke versi 3. 12 dianggap sebagai upgrade versi minor.

Anda dapat meningkatkan broker Anda secara manual kapan saja ke versi mayor atau minor yang didukung berikutnya. Amazon MQ mengelola peningkatan ke versi patch terbaru yang didukung untuk semua broker selama jendela pemeliharaan terjadwal. Upgrade versi manual dan otomatis terjadi selama jendela pemeliharaan terjadwal, atau setelah Anda me-reboot broker Anda. Amazon MQ meningkatkan broker Anda ke versi minor berikutnya ketika versi minor saat ini mencapai akhir dukungan.

Meningkatkan versi mesin secara manual

Anda dapat memutakhirkan versi mesin broker dengan menggunakan Konsol Manajemen AWS, API AWS CLI, atau Amazon MQ.

Konsol Manajemen AWS

Untuk meng-upgrade versi mesin broker dengan menggunakan Konsol Manajemen AWS

1. Pada halaman detail broker, pilih Edit.
2. Di bawah Spesifikasi, untuk Versi mesin broker, pilih nomor versi baru dari daftar dropdown.
3. Gulir ke bagian bawah halaman, lalu pilih Jadwalkan perubahan.
4. Di halaman Jadwalkan modifikasi broker, untuk Kapan menerapkan perubahan, pilih salah satu opsi berikut.
   - Pilih Setelah boot ulang berikutnya, jika Anda ingin Amazon MQ untuk menyelesaikan peningkatan versi selama jendela pemeliharaan terjadwal berikutnya.
   - Pilih Segera, jika Anda ingin segera melakukan boot ulang broker dan meningkatkan versi mesin.

   > **Important**
   >
   > Pialang instans tunggal sedang offline saat di-boot ulang. Untuk broker cluster, hanya satu node yang turun pada satu waktu sementara broker melakukan reboot.

5. Pilih Terapkan untuk menyelesaikan penerapan perubahan.

AWS CLI

Untuk meng-upgrade versi mesin broker dengan menggunakan AWS CLI

1. Gunakan perintah CLI update-broker dan tentukan parameter berikut, seperti yang ditampilkan dalam contoh.
   - `--broker-id` – ID unik yang dihasilkan Amazon MQ untuk broker. Anda dapat mengurai ID dari ARN broker. Misalnya, dengan ARN berikut, `arn:aws:mq:us-east-2:123456789012:broker:MyBroker:b-1234a5b6-78cd-901e-2fgh-3i45j6k178l9`, ID broker akan menjadi `b-1234a5b6-78cd-901e-2fgh-3i45j6k178l9`.
   - `--engine-version` – Nomor versi mesin untuk meningkatkan mesin broker.

```
aws mq update-broker --broker-id broker-id --engine-version version-number
```

2. (Opsional) Gunakan perintah CLI reboot-broker untuk me-reboot broker Anda jika Anda ingin segera memutakhirkan versi mesin.

```
aws mq reboot-broker --broker-id broker-id
```

Jika Anda tidak ingin melakukan boot ulang broker dan segera menerapkan perubahan, Amazon MQ akan meningkatkan broker selama jendela pemeliharaan terjadwal berikutnya.

> **Important**
>
> Pialang instans tunggal sedang offline saat di-boot ulang. Untuk broker cluster, hanya satu node yang turun pada satu waktu sementara broker melakukan reboot.

API Amazon MQ

Untuk meningkatkan versi mesin broker menggunakan API Amazon MQ

1. Gunakan Operasi API UpdateBroker. Tentukan `broker-id` sebagai parameter jalur. Contoh berikut mengasumsikan broker di wilayah `us-west-2`. Untuk informasi selengkapnya tentang titik akhir Amazon MQ yang tersedia, lihat titik akhir dan kuota Amazon MQ. di Referensi Umum AWS

```
PUT /v1/brokers/broker-id HTTP/1.1
Host: mq.us-west-2.amazonaws.com
Date: Mon, 7 June 2021 12:00:00 GMT
x-amz-date: Mon, 7 June 2021 12:00:00 GMT
Authorization: authorization-string
```

Gunakan `engineVersion` dalam muatan permintaan untuk menentukan nomor versi peningkatan broker.

```
{
    "engineVersion": "engine-version-number"
}
```

2. (Opsional) Gunakan operasi RebootBrokerAPI untuk me-reboot broker Anda jika Anda ingin segera memutakhirkan versi mesin. `broker-id`ditentukan sebagai parameter jalur.

```
POST /v1/brokers/broker-id/reboot-broker HTTP/1.1
Host: mq.us-west-2.amazonaws.com
Date: Mon, 7 June 2021 12:00:00 GMT
x-amz-date: Mon, 7 June 2021 12:00:00 GMT
Authorization: authorization-string
```

Jika Anda tidak ingin melakukan boot ulang broker dan segera menerapkan perubahan, Amazon MQ akan meningkatkan broker selama jendela pemeliharaan terjadwal berikutnya.

> **Important**
>
> Pialang instans tunggal sedang offline saat di-boot ulang. Untuk broker cluster, hanya satu node yang turun pada satu waktu sementara broker melakukan reboot.

Memutakhirkan jenis instans broker Amazon MQ

> **Important**
>
> `mq.m7g.x`instance hanya tersedia untuk Amazon MQ untuk broker RabbitMQ. Amazon MQ untuk broker ActiveMQ hanya menggunakan instance. `mq.m5.x`

Deskripsi gabungan dari kelas instance broker (`m7g`) dan size (`large`) disebut tipe instance broker (misalnya,`mq.m7g.large`). Saat memilih jenis instans, penting untuk mempertimbangkan faktor-faktor yang akan mempengaruhi kinerja broker:

- jumlah klien dan antrian
- volume pesan yang dikirim
- pesan disimpan dalam memori
- pesan berlebihan

Jenis instans broker yang lebih kecil (`mq.m7g.medium`) direkomendasikan hanya untuk menguji kinerja aplikasi. Kami merekomendasikan jenis instans broker yang lebih besar (`mq.m7g.large` dan di atasnya) untuk tingkat produksi klien dan antrian, throughput tinggi, pesan dalam memori, dan pesan yang berlebihan.

Sebaiknya upgrade ke jenis instans yang lebih besar (yaitu dari `micro` ke `large`) jika Anda mengalami masalah kinerja, atau jika Anda beralih dari pengujian ke lingkungan produksi. Untuk memutakhirkan jenis instans, Anda dapat menggunakan Konsol Manajemen AWS API MQ Amazon AWS CLI, atau Amazon MQ.

Konsol Manajemen AWS

Untuk meningkatkan ke jenis instans yang lebih besar menggunakan Konsol Manajemen AWS, lakukan hal berikut:

1. Masuk ke konsol Amazon MQ.
2. Pada panel navigasi kiri, pilih Broker, lalu pilih broker dalam daftar yang ingin Anda tingkatkan.
3. Pada halaman detail broker, pilih Edit.
4. Di bawah Spesifikasi, untuk jenis instans Broker pilih jenis instans baru dari daftar dropdown.
5. Di bagian bawah halaman, pilih Jadwalkan modifikasi.
6. Di halaman Jadwalkan modifikasi broker, untuk Kapan menerapkan perubahan, pilih salah satu opsi berikut.

   - Pilih Setelah reboot berikutnya, jika Anda ingin Amazon MQ menyelesaikan peningkatan selama jendela pemeliharaan terjadwal berikutnya.
   - Pilih Segera, jika Anda ingin me-reboot broker dan segera memutakhirkan jenis instans.

   > **Important**
   >
   > Pialang instans tunggal sedang offline saat di-boot ulang. Untuk broker cluster, hanya satu node yang turun pada satu waktu sementara broker melakukan reboot.

7. Pilih Terapkan untuk menyelesaikan penerapan perubahan.

AWS CLI

Untuk meng-upgrade jenis instans broker dengan menggunakan AWS CLI

1. Gunakan perintah CLI modify-broker dan tentukan parameter berikut, seperti yang ditunjukkan pada contoh.

   - `--broker-id` – ID unik yang dihasilkan Amazon MQ untuk broker.

- `--host-instance-type` – Nomor versi mesin untuk meningkatkan mesin broker.

```
aws mq modify-broker --broker-id broker-id --host-instance-type instance-type
```

2. (Opsional) Gunakan perintah CLI reboot-broker untuk me-reboot broker Anda jika, Anda ingin segera memutakhirkan jenis instance.

```
aws mq reboot-broker --broker-id broker-id
```

Jika Anda tidak ingin melakukan boot ulang broker dan segera menerapkan perubahan, Amazon MQ akan meningkatkan broker selama jendela pemeliharaan terjadwal berikutnya.

> **Important**
>
> Pialang instans tunggal sedang offline saat di-boot ulang. Untuk broker cluster, hanya satu node yang turun pada satu waktu sementara broker melakukan reboot.

API Amazon MQ

Untuk memutakhirkan tipe instans broker dengan menggunakan Amazon MQ API

1. Gunakan Operasi API UpdateBroker. Tentukan `broker-id` sebagai parameter jalur. Contoh berikut mengasumsikan broker di wilayah `us-west-2`. Untuk informasi selengkapnya tentang titik akhir Amazon MQ yang tersedia, lihat titik akhir dan kuota Amazon MQ di. Referensi Umum AWS

```
PUT /v1/brokers/broker-id HTTP/1.1
Host: mq.us-west-2.amazonaws.com
Date: Mon, 7 June 2021 12:00:00 GMT
x-amz-date: Mon, 7 June 2021 12:00:00 GMT
Authorization: authorization-string
```

Gunakan `host-instance-type` dalam payload permintaan untuk menentukan jenis instans untuk broker untuk meningkatkan ke.

```
{
    "host-instance-type": "host-instance-type"
```

```
}
```

2. (Opsional) Gunakan operasi RebootBrokerAPI untuk me-reboot broker Anda, jika Anda ingin segera memutakhirkan versi mesin. `broker-id`ditentukan sebagai parameter jalur.

```
POST /v1/brokers/broker-id/reboot-broker HTTP/1.1
Host: mq.us-west-2.amazonaws.com
Date: Mon, 7 June 2021 12:00:00 GMT
x-amz-date: Mon, 7 June 2021 12:00:00 GMT
Authorization: authorization-string
```

Jika Anda tidak ingin melakukan boot ulang broker dan segera menerapkan perubahan, Amazon MQ akan meningkatkan broker selama jendela pemeliharaan terjadwal berikutnya.

> **Important**
>
> Pialang instans tunggal sedang offline saat di-boot ulang. Untuk broker cluster, hanya satu node yang turun pada satu waktu sementara broker melakukan reboot.

Amazon MQ untuk jenis penyimpanan ActiveMQ

Amazon MQ for ActiveMQ mendukung Amazon Elastic File System (EFS) dan Amazon Elastic Block Store (EBS). Secara default, broker ActiveMQ menggunakan Amazon EFS untuk penyimpanan broker. Untuk memanfaatkan daya tahan dan replikasi yang tinggi di beberapa Availability Zone, gunakan Amazon EFS. Untuk memanfaatkan latensi rendah dan throughput yang tinggi, gunakan Amazon EBS.

> **Important**
>
> - Anda dapat menggunakan Amazon EBS hanya dengan keluarga tipe instans broker `mq.m5`.
> - Meski Anda dapat mengubah tipe instans broker, Anda tidak dapat mengubah tipe penyimpanan broker setelah Anda membuat broker.
> - Amazon EBS mereplikasi data dalam satu Availability Zone dan tidak mendukung mode deployment ActiveMQ aktif/siaga.

Perbedaan beragam Jenis Penyimpanan

Tabel berikut memberikan gambaran umum singkat tentang perbedaan antara jenis penyimpanan dalam memori, Amazon EFS, dan Amazon EBS untuk broker ActiveMQ.

Jenis Penyimpanan	Tetap	Contoh Kasus Penggunaan	Perkiraan Jumlah Pesan Maksimum yang Dieantrekan per Produsen, per detik (Pesan 1KB)	Replikasi
Dalam memori	Tidak tetap	• Tanda kutip saham • Pembaruan data lokasi • Data yang sering diubah	5.000	Tidak ada
Amazon EBS	Tetap	• Volume teks yang tinggi • Proses pemesanan	500	Beberapa salinan dalam satu Availability Zone (AZ)
Amazon EFS	Tetap	Transaksi keuangan	80	Beberapa salinan di beberapa AZs

Penyimpanan pesan dalam memori memberikan latensi paling rendah dan throughput paling tinggi. Namun, pesan hilang selama penggantian instans atau mulai ulang broker.

Amazon EFS dirancang agar sangat tahan lama, direplikasi di beberapa AZs untuk mencegah hilangnya data akibat kegagalan komponen tunggal atau masalah yang memengaruhi ketersediaan AZ. Amazon EBS dioptimalkan untuk throughput dan direplikasi di beberapa server dalam AZ tunggal.

Mengkonfigurasi broker MQ Amazon pribadi

Pialang pribadi tidak memiliki aksesibilitas publik dan tidak dapat diakses dari luar VPC Anda. Sebelum mengonfigurasi broker pribadi, lihat informasi berikut tentang VPCs, subnet, dan grup keamanan:

- VPCs
  - Subnet broker dan grup keamanan harus berada dalam VPC yang sama.
  - Saat Anda menggunakan broker pribadi, Anda mungkin melihat alamat IP yang tidak Anda konfigurasikan dengan VPC Anda. Ini adalah alamat IP dari infrastruktur Amazon MQ, dan mereka tidak memerlukan tindakan.
- Subnet
  - Jika subnet berada dalam VPC bersama, VPC harus dimiliki oleh akun yang sama yang membuat broker.
  - Jika tidak ada subnet yang disediakan, subnet default di VPC default akan digunakan.
  - Setelah broker dibuat, subnet yang digunakan tidak dapat diubah.
  - Untuk cluster dan active/standby broker, subnet harus berada di Availability Zone yang berbeda.
  - Untuk broker contoh tunggal, Anda dapat menentukan subnet mana yang akan digunakan dan broker akan dibuat dalam Availability Zone yang sama.
- Grup keamanan
  - Jika tidak ada grup keamanan yang disediakan, grup keamanan default di VPC default akan digunakan.
  - Single-instance, cluster, dan active/standby broker memerlukan setidaknya satu grup keamanan (misalnya, grup keamanan default).

    > **Note**
    >
    > Broker RabbitMQ publik tidak menggunakan subnet atau grup keamanan.

  - Setelah broker dibuat, grup keamanan yang digunakan tidak dapat diubah. Kelompok keamanan sendiri masih dapat dimodifikasi.

Mengkonfigurasi broker pribadi di Konsol Manajemen AWS

Untuk mengkonfigurasi broker pribadi, mulailah membuat broker baru di Konsol Manajemen AWS. Kemudian, di bagian Pengaturan jaringan, untuk mengonfigurasi konektivitas broker Anda, lakukan hal berikut:

1. Pilih akses pribadi untuk broker Anda. Untuk terhubung ke broker pribadi, Anda dapat menggunakan IPv4, IPv6, atau dual-stack (IPv4 dan IPv6). Untuk informasi selengkapnya, lihat Connecting to Amazon MQ.
2. Selanjutnya, pilih Gunakan VPC default, subnet, dan grup keamanan, atau pilih Pilih VPC, subnet, dan grup keamanan yang ada. Jika Anda tidak ingin menggunakan VPC, subnet, atau grup keamanan default atau yang sudah ada, Anda harus membuat yang baru untuk terhubung ke broker pribadi.

> **Note**
>
> Untuk akses broker pribadi, metode koneksi akan sama dengan jenis IP subnet yang dipilih. Setelah broker dibuat, titik akhir VPC tidak dapat diubah dan akan selalu memiliki jenis IP dari subnet yang dipilih. Jika Anda ingin menggunakan jenis IP baru, Anda harus membuat broker baru.

> **Note**
>
> Amazon MQ untuk ActiveMQ tidak menggunakan titik akhir VPC. Saat pertama kali membuat broker ActiveMQ, Amazon MQ menyediakan elastic network interface (ENI) di VPC. Kelompok keamanan ditempatkan di ENI dan dapat digunakan untuk broker publik dan swasta.

Mengakses konsol web broker Amazon MQ tanpa aksesibilitas publik

Ketika Anda mematikan aksesibilitas publik untuk broker Anda, ID AWS akun yang membuat broker dapat mengakses broker pribadi. Jika Anda mematikan aksesibilitas publik untuk broker Anda, Anda harus melakukan langkah-langkah berikut untuk mengakses konsol web broker.

1. Buat instans EC2 Linux di `public-vpc` (dengan IP publik, jika perlu).

2. Untuk memverifikasi bahwa VPC Anda dikonfigurasi dengan benar, buat koneksi `ssh` ke instans EC2 dan gunakan perintah `curl` dengan URI broker Anda.
3. Dari mesin Anda, buat terowongan `ssh` ke instans EC2 menggunakan jalur ke file kunci privat dan alamat IP instans EC2 publik Anda. Sebagai contoh:

   ```
   ssh -i ~/.ssh/id_rsa -N -C -q -f -D 8080 ec2-user@203.0.113.0
   ```

   Server proksi terusan dimulai pada mesin Anda.
4. Instal klien proxy seperti FoxyProxydi mesin Anda.
5. Konfigurasikan klien proksi menggunakan pengaturan berikut:

   - Untuk tipe proksi, tentukan `SOCKS5`.
   - Untuk alamat IP, nama DNS, dan nama server, tentukan `localhost`.
   - Untuk port, tentukan `8080`.
   - Menghapus pola URL yang ada.
   - Untuk pola URL, tentukan `*.mq.*.amazonaws.com*`
   - Untuk jenis koneksi, tentukan `HTTP(S)`.

   Jika klien proksi diaktifkan, Anda dapat mengakses konsol web di mesin Anda.

> **Important**
>
> Jika Anda menggunakan broker pribadi, Anda mungkin melihat alamat IP yang tidak Anda konfigurasikan dengan VPC Anda. Ini adalah alamat IP dari RabbitMQ di infrastruktur Amazon MQ, dan mereka tidak memerlukan tindakan.

Menjadwalkan jendela pemeliharaan untuk broker Amazon MQ

Secara berkala, Amazon MQ melakukan pemeliharaan pada perangkat keras, sistem operasi, atau perangkat lunak mesin dari broker pesan selama jendela pemeliharaan. Misalnya, jika Anda mengubah jenis instans broker, Amazon MQ akan menerapkan perubahan Anda selama jendela pemeliharaan terjadwal berikutnya. Durasi pemeliharaan dapat bertahan hingga dua jam tergantung pada operasi yang dijadwalkan untuk broker pesan Anda. Anda dapat meminimalkan waktu henti

selama jendela pemeliharaan dengan memilih mode penyebaran broker dengan ketersediaan tinggi di beberapa Availability Zone (AZ).

Amazon MQ untuk ActiveMQ menyediakan penerapan aktif/siaga untuk ketersediaan tinggi. Dalam active/standby mode, Amazon MQ melakukan operasi pemeliharaan satu instance pada satu waktu, dan setidaknya satu instance tetap tersedia. Selain itu, Anda dapat mengonfigurasi jaringan broker dengan jendela pemeliharaan bervariasi sepanjang minggu. Amazon MQ untuk RabbitMQ menyediakan penerapan cluster untuk ketersediaan tinggi. Dalam penerapan cluster, Amazon MQ melakukan operasi pemeliharaan satu node pada satu waktu dengan menjaga setidaknya dua node yang berjalan setiap saat.

Ketika Anda pertama kali membuat broker Anda, Anda dapat menjadwalkan jendela pemeliharaan terjadi seminggu sekali pada waktu yang ditentukan. Anda hanya dapat menyesuaikan jendela pemeliharaan broker hingga empat kali sebelum jendela pemeliharaan terjadwal berikutnya. Setelah jendela pemeliharaan broker selesai, Amazon MQ mengatur ulang batas, dan Anda dapat menyesuaikan jadwal lagi sebelum jendela pemeliharaan berikutnya terjadi. Ketersediaan broker tidak terpengaruh saat menyesuaikan jendela pemeliharaan broker.

Untuk menyesuaikan jendela pemeliharaan broker, Anda dapat menggunakan Konsol Manajemen AWS, AWS CLI, atau Amazon MQ API.

Jadwalkan jendela pemeliharaan broker menggunakan Konsol Manajemen AWS

Untuk menyesuaikan jendela pemeliharaan broker dengan menggunakan Konsol Manajemen AWS

1. Masuk ke konsol Amazon MQ.
2. Pada panel navigasi kiri, pilih Broker, lalu pilih broker dalam daftar yang ingin Anda tingkatkan.
3. Pada halaman detail broker, pilih Edit.
4. Dalam Pemeliharaan, lakukan hal berikut.
    a. Untuk Hari mulai, pilih hari dalam seminggu, misalnya, hari Minggu, dari daftar drop-down.
    b. Untuk waktu Mulai, pilih jam dan menit hari yang ingin Anda jadwalkan untuk jendela pemeliharaan broker berikutnya, misalnya, 12: 00.

> **Note**
>
> Opsi waktu mulai dikonfigurasi dalam zona waktu UTC+0.

5. Selanjutnya, pilih Jadwalkan modifikasi. Kemudian pilih Setelah reboot berikutnya atau Segera. Memilih Setelah reboot berikutnya akan segera memperbarui jendela pemeliharaan tanpa me-reboot broker. Memilih Segera akan segera reboot broker.
6. Pada halaman detail broker, di bawah jendela Pemeliharaan, verifikasi bahwa jadwal pilihan baru Anda ditampilkan.

Jadwalkan jendela pemeliharaan broker menggunakan AWS CLI

Untuk menyesuaikan jendela pemeliharaan broker menggunakan AWS CLI

1. Gunakan perintah CLI update-broker dan tentukan parameter berikut, seperti yang ditampilkan dalam contoh.

   - `--broker-id` – ID unik yang dihasilkan Amazon MQ untuk broker. Anda dapat mengurai ID dari ARN broker. Misalnya, dengan ARN berikut, `arn:aws:mq:us-east-2:123456789012:broker:MyBroker:b-1234a5b6-78cd-901e-2fgh-3i45j6k178l9`, ID broker akan menjadi `b-1234a5b6-78cd-901e-2fgh-3i45j6k178l9`.
   - `--maintenance-window-start-time`— Parameter yang menentukan waktu mulai jendela pemeliharaan mingguan yang disediakan dalam struktur berikut.
     - `DayOfWeek`— Hari dalam seminggu, dalam sintaks berikut: `MONDAY| TUESDAY | WEDNESDAY | THURSDAY | FRIDAY | SATURDAY | SUNDAY`
     - `TimeOfDay`— Waktu, dalam format 24 jam.
     - `TimeZone`— (Opsional) Zona waktu, baik dalam Negara/Kota, atau format offset UTC. Setel ke UTC secara default.

   ```
   aws mq update-broker --broker-id broker-id \
   --maintenance-window-start-time DayOfWeek=SUNDAY,TimeOfDay=13:00,TimeZone=America/Los_Angeles
   ```

2. (Opsional) Gunakan perintah CLI deskripsi-broker untuk memverifikasi bahwa jendela pemeliharaan berhasil diperbarui.

   ```
   aws mq describe-broker --broker-id broker-id
   ```

Jadwalkan jendela pemeliharaan broker menggunakan Amazon MQ API

Untuk menyesuaikan jendela pemeliharaan broker menggunakan Amazon MQ API

1. Gunakan Operasi API UpdateBroker. Tentukan `broker-id` sebagai parameter jalur. Contoh berikut mengasumsikan broker di wilayah `us-west-2`. Untuk informasi selengkapnya tentang titik akhir Amazon MQ yang tersedia, lihat titik akhir dan kuota Amazon MQ di. Referensi Umum AWS

   ```
   PUT /v1/brokers/broker-id HTTP/1.1
   Host: mq.us-west-2.amazonaws.com
   Date: Wed, 7 July 2021 12:00:00 GMT
   x-amz-date: Wed, 7 July 2021 12:00:00 GMT
   Authorization: authorization-string
   ```

   Gunakan `maintenanceWindowStartTime` parameter dan jenis `WeeklyStartTime`sumber daya dalam payload permintaan.

   ```
   {
   "maintenanceWindowStartTime": {
       "dayOfWeek": "SUNDAY",
       "timeZone": "America/Los_Angeles",
       "timeOfDay": "13:00"
     }
   }
   ```

2. (Opsional) Gunakan operasi DescribeBrokerAPI untuk memverifikasi bahwa jendela pemeliharaan telah berhasil diperbarui. `broker-id`ditentukan sebagai parameter jalur.

   ```
   GET /v1/brokers/broker-id HTTP/1.1
   Host: mq.us-west-2.amazonaws.com
   Date: Wed, 7 July 2021 12:00:00 GMT
   x-amz-date: Wed, 7 July 2021 12:00:00 GMT
   Authorization: authorization-string
   ```

Melakukan boot ulang broker Amazon MQ

Untuk menerapkan konfigurasi baru ke broker, Anda dapat melakukan boot ulang broker.

> **Note**
>
> Jika broker ActiveMQ Anda menjadi tidak responsif, Anda dapat mem-boot ulang untuk memulihkan dari keadaan yang salah.

Contoh berikut menunjukkan cara melakukan boot ulang broker Amazon MQ menggunakan Konsol Manajemen AWS.

Untuk Melakukan Boot Ulang Broker Amazon MQ

1. Masuk ke konsol Amazon MQ.
2. Dari daftar broker, pilih nama broker Anda (misalnya, MyBroker).
3. Pada ***MyBroker***halaman, pilih Actions, Reboot broker.

   > **Important**
   >
   > Pialang instans tunggal akan offline saat di-boot ulang. Broker cluster akan tersedia, tetapi setiap node di-reboot satu per satu.

4. Di kotak dialaog Boot ulang broker, pilih Boot ulang.

   Reboot broker membutuhkan waktu sekitar 5 menit. Jika reboot menyertakan perubahan ukuran instance atau dilakukan pada broker dengan kedalaman antrian tinggi, proses reboot bisa memakan waktu lebih lama.

Menghapus broker Amazon MQ

Jika Anda tidak menggunakan broker Amazon MQ (dan tidak memperkirakan menggunakannya dalam waktu dekat), itu adalah praktik terbaik untuk menghapusnya dari Amazon MQ untuk mengurangi biaya Anda. AWS

Contoh berikut menunjukkan cara menghapus broker menggunakan Konsol Manajemen AWS.

Menghapus broker Amazon MQ

1. Masuk ke konsol Amazon MQ.

2. Dari daftar broker, pilih broker Anda (misalnya, MyBroker) dan kemudian pilih Hapus.
3. Di Hapus***MyBroker***? kotak dialog, ketik `delete` dan kemudian pilih Hapus.

   Menghapus broker berlangsung sekitar 5 menit.

Status broker Amazon MQ

Kondisi broker saat ini ditunjukkan dengan status. Tabel berikut mencantumkan status broker Amazon MQ.

Konsol	API	Deskripsi
Pembuatan gagal	`CREATION_FAILED`	Broker tidak dapat dibuat.
Pembuatan sedang berlangsung	`CREATION_IN_PROGRESS`	Saat ini broker sedang dibuat.
Penghapusan sedang berlangsung	`DELETION_IN_PROGRESS`	Saat ini broker sedang dihapus.
Boot ulang sedang berlangsung	`REBOOT_IN_PROGRESS`	Saat ini broker sedang di-boot ulang.
Berjalan	`RUNNING`	Broker dapat dioperasikan.
Tindakan kritis diperlukan	`CRITICAL_ACTION_REQUIRED`	Pialang sedang berjalan, tetapi dalam keadaan terdegradasi dan membutuhkan tindakan segera. Anda dapat menemukan petunjuk untuk menyelesaikan masalah dengan memilih kode tindakan yang diperlukan dari daftar diPemecahan masalah.

Menambahkan tag ke sumber daya Amazon MQ

Untuk mengelola dan mengidentifikasi sumber daya Amazon MQ untuk alokasi biaya, Anda dapat menambahkan tanda metadata yang mengidentifikasi tujuan broker atau konfigurasi. Ini sangat berguna jika Anda memiliki banyak broker. Anda dapat menggunakan tag alokasi biaya untuk mengatur AWS tagihan Anda untuk mencerminkan struktur biaya Anda sendiri. Untuk melakukan ini, daftar untuk mendapatkan tagihan AWS akun Anda untuk menyertakan kunci tag dan nilai. Untuk informasi selengkapnya, lihat Menyiapkan Laporan Alokasi Biaya Bulanan dalam Panduan Pengguna AWS Billing .

Misalnya, Anda dapat menambahkan tanda yang mewakili pusat biaya dan tujuan sumber daya Amazon MQ:

Sumber Daya	Kunci	Nilai
`Broker1`	`Cost Center`	`34567`
	`Stack`	`Production`
`Broker2`	`Cost Center`	`34567`
	`Stack`	`Production`
`Broker3`	`Cost Center`	`12345`
	`Stack`	`Development`

Skema penandaan ini memungkinkan Anda mengelompokkan dua broker yang melakukan tugas terkait di pusat biaya yang sama, seraya menandai broker yang tidak terkait dengan tanda alokasi biaya yang berbeda.

Menambahkan tag di Amazon MQ Console

Anda dapat dengan cepat menambahkan tag ke sumber daya yang Anda buat di konsol Amazon MQ dengan mengikuti langkah-langkah berikut:

1. Dari halaman Buat broker, pilih Pengaturan tambahan.
2. Di bawah Tanda, pilih Tambah tanda.

3. Masukkan pasangan Kunci dan Nilai.
4. (Opsional) Pilih Tambah tanda untuk menambahkan beberapa tanda ke broker Anda.
5. Pilih Buat broker.

Untuk menambahkan tanda saat Anda membuat konfigurasi:

1. Dari halaman Buat Konfigurasi, pilih Lanjutan.
2. Di bawah Tanda pada halaman Buat konfigurasi, pilih Tambah tanda.
3. Masukkan pasangan Kunci dan Nilai.
4. (Opsional) Pilih Tambah tanda untuk menambahkan beberapa tanda ke konfigurasi Anda.
5. Pilih Buat konfigurasi.

Setelah menambahkan tag, Anda dapat melihat, mengedit, dan menghapus tag untuk sumber daya Anda di konsol Amazon MQ. Anda juga dapat melihat tag sumber daya Anda menggunakan REST API. Untuk informasi selengkapnya, lihat Referensi REST API Amazon MQ.

Menggunakan Amazon MQ untuk ActiveMQ

Amazon MQ memudahkan pembuatan broker pesan dengan sumber daya komputasi dan penyimpanan yang sesuai dengan kebutuhan Anda. Anda dapat membuat, mengelola, dan menghapus broker menggunakan Konsol Manajemen AWS, Amazon MQ REST API, atau. AWS Command Line Interface

Amazon MQ untuk broker ActiveMQ dapat digunakan sebagai broker instans tunggal atau pialang aktif/siaga. Untuk kedua mode deployment, Amazon MQ memberikan daya tahan tinggi dengan menyimpan data secara redundan.

> **Note**
>
> Amazon MQ menggunakan Apache KahaDB sebagai penyimpanan data. Penyimpanan data lainnya, seperti JDBC dan LevelDB, tidak didukung.

Anda dapat mengakses broker menggunakan bahasa pemrograman yang didukung ActiveMQ dan dengan mengaktifkan TLS secara eksplisit untuk protokol berikut:

- AMQP
- MQTT
- MQTT lebih WebSocket
- OpenWire
- MENGINJAK
- STOMP berakhir WebSocket

Untuk mempelajari tentang Amazon MQ REST APIs, lihat Referensi API Amazon MQ REST.

Amazon MQ untuk broker ActiveMQ

Apa itu Amazon MQ untuk broker ActiveMQ?

Broker adalah lingkungan broker pesan yang berjalan di Amazon MQ. Ini adalah blok bangunan dasar Amazon MQ. Deskripsi gabungan dari kelas instance broker (`m5`) dan size (`large`,`medium`)

disebut tipe instance broker (misalnya,`mq.m5.large`). Untuk informasi selengkapnya, lihat Broker instance types.

- Broker single instance terdiri dari satu broker dalam satu Availability Zone. Broker berkomunikasi dengan aplikasi Anda dan dengan volume penyimpanan Amazon EBS atau Amazon EFS.
- Broker aktif/siaga terdiri dari dua broker di dua Availability Zone yang berbeda, dikonfigurasi dalam pasangan redundan. Broker ini berkomunikasi secara sinkron dengan aplikasi Anda dan Amazon EFS.

Untuk informasi selengkapnya, lihat Opsi penyebaran untuk Amazon MQ untuk broker ActiveMQ.

Anda dapat mengaktifkan peningkatan versi minor otomatis ke versi minor baru dari mesin broker, karena Apache merilis versi baru. Peningkatan otomatis terjadi selama jendela pemeliharaan yang ditentukan oleh hari dalam seminggu, waktu dalam sehari (dalam format 24 jam), dan zona waktu (UTC secara default).

Untuk informasi tentang membuat dan mengelola broker, lihat hal berikut:

- Memulai: Membuat dan menghubungkan ke broker ActiveMQ
- Pialang
- Broker statuses

Protokol tingkat wire yang didukung

Anda dapat mengakses broker menggunakan bahasa pemrograman yang didukung ActiveMQ dan dengan mengaktifkan TLS secara eksplisit untuk protokol berikut:

- AMQP
- MQTT
- MQTT lebih WebSocket
- OpenWire
- MENGINJAK
- STOMP berakhir WebSocket

Atribut

Broker ActiveMQ memiliki beberapa atribut, misalnya:

- Nama (`MyBroker`)
- ID (`b-1234a5b6-78cd-901e-2fgh-3i45j6k178l9`)
- Amazon Resource Name (ARN) (`arn:aws:mq:us-east-2:123456789012:broker:MyBroker:b-1234a5b6-78cd-901e-2fgh-3i45j6k178l9`)
- URL Konsol Web ActiveMQ (`https://b-1234a5b6-78cd-901e-2fgh-3i45j6k178l9-1.mq.us-east-2.amazonaws.com:8162`)

  Untuk informasi selengkapnya, lihat Konsol Web dalam dokumentasi Apache ActiveMQ.

  > **Important**
  >
  > Jika menentukan peta otorisasi yang tidak menyertakan grup `activemq-webconsole`, Anda tidak dapat menggunakan Konsol Web ActiveMQ karena grup tidak berwenang untuk mengirim pesan ke, atau menerima pesan dari, broker Amazon MQ.

- Titik akhir protokol tingkat wire:
  - `amqp+ssl://b-1234a5b6-78cd-901e-2fgh-3i45j6k178l9-1.mq.us-east-2.amazonaws.com:5671`
  - `mqtt+ssl://b-1234a5b6-78cd-901e-2fgh-3i45j6k178l9-1.mq.us-east-2.amazonaws.com:8883`
  - `ssl://b-1234a5b6-78cd-901e-2fgh-3i45j6k178l9-1.mq.us-east-2.amazonaws.com:61617`

    > **Note**
    >
    > Ini adalah OpenWire titik akhir.

  - `stomp+ssl://b-1234a5b6-78cd-901e-2fgh-3i45j6k178l9-1.mq.us-east-2.amazonaws.com:61614`
  - `wss://b-1234a5b6-78cd-901e-2fgh-3i45j6k178l9-1.mq.us-east-2.amazonaws.com:61619`

Untuk informasi selengkapnya, lihat Mengonfigurasi Transportasi dalam dokumentasi Apache ActiveMQ.

> **Note**
>
> Untuk active/standby broker, Amazon MQ menyediakan dua ActiveMQ Web Console URLs, tetapi hanya satu URL yang aktif pada satu waktu. Demikian juga, Amazon MQ menyediakan dua titik akhir untuk setiap protokol tingkat wire, tetapi hanya satu titik akhir aktif di setiap pasangan pada satu waktu. Sufiks `-1` dan `-2` menunjukkan pasangan redundan.

Untuk daftar lengkap atribut broker, lihat di Referensi REST API Amazon MQ:

- ID Operasi REST: Broker
- ID Operasi REST: Broker
- ID Operasi REST: Reboot Broker

Pengguna broker

Pengguna ActiveMQ adalah orang atau aplikasi yang dapat mengakses antrean dan topik broker ActiveMQ. Anda dapat mengonfigurasi pengguna untuk memiliki izin tertentu. Misalnya, Anda dapat mengizinkan beberapa pengguna mengakses Konsol Web ActiveMQ.

Grup adalah label semantik. Anda dapat menetapkan grup ke pengguna dan mengonfigurasi izin untuk grup untuk mengikirim ke, menerima dari, dan mengelola antrean serta topik tertentu.

> **Important**
>
> Pembuatan perubahan pada pengguna tidak akan segera menerapkan perubahan ke pengguna. Untuk menerapkan perubahan Anda, Anda harus menunggu jendela pemeliharaan berikutnya atau reboot broker.

Untuk informasi tentang pengguna dan grup, lihat hal berikut dalam dokumentasi Apache ActiveMQ:

- Otorisasi
- Contoh Otorisasi

Untuk informasi tentang membuat, mengedit, dan menghapus pengguna ActiveMQ, lihat hal berikut:

- Membuat pengguna broker ActiveMQ
- Pengguna

Atribut pengguna

Untuk daftar lengkap atribut pengguna, lihat di Referensi REST API Amazon MQ:

- ID Operasi REST: Pengguna
- ID Operasi REST: Pengguna

Opsi penyebaran untuk Amazon MQ untuk broker ActiveMQ

Amazon MQ menawarkan opsi penyebaran instance dan cluster tunggal untuk broker.

Opsi 1: Pialang instans tunggal Amazon MQ

Broker single instance terdiri dari satu broker dalam satu Availability Zone. Broker berkomunikasi dengan aplikasi Anda dan dengan volume penyimpanan Amazon EBS atau Amazon EFS. Volume penyimpanan Amazon EFS dirancang untuk memberikan tingkat daya tahan dan ketersediaan tertinggi dengan menyimpan data secara berlebihan di beberapa Availability Zone ()AZs. Amazon EBS menyediakan penyimpanan tingkat blok yang dioptimalkan untuk latensi rendah dan throughput tinggi. Untuk informasi selengkapnya tentang opsi penyimpanan, lihatStorage.

Diagram berikut mengilustrasikan broker instans tunggal dengan penyimpanan Amazon EFS yang direplikasi di beberapa. AZs

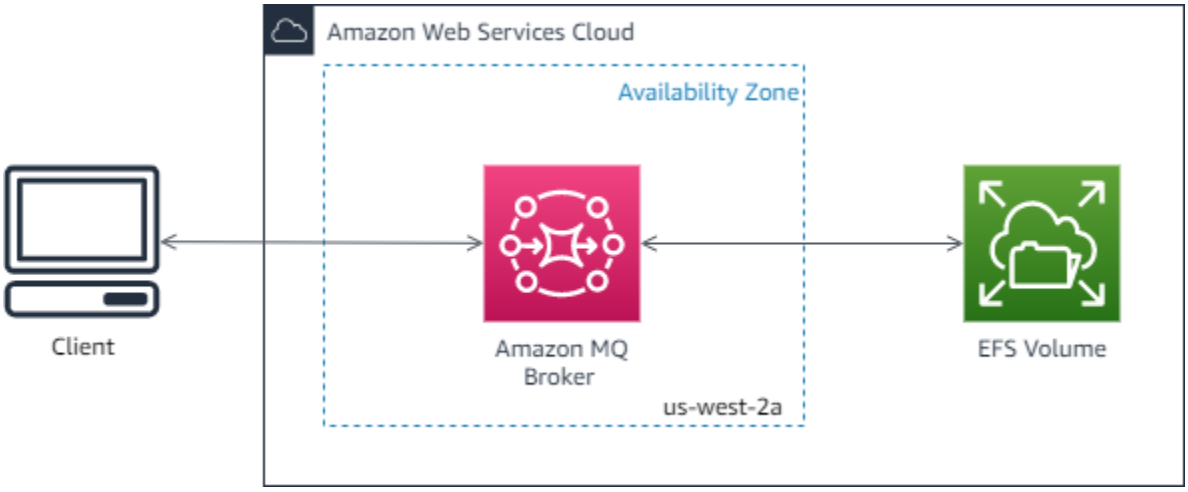
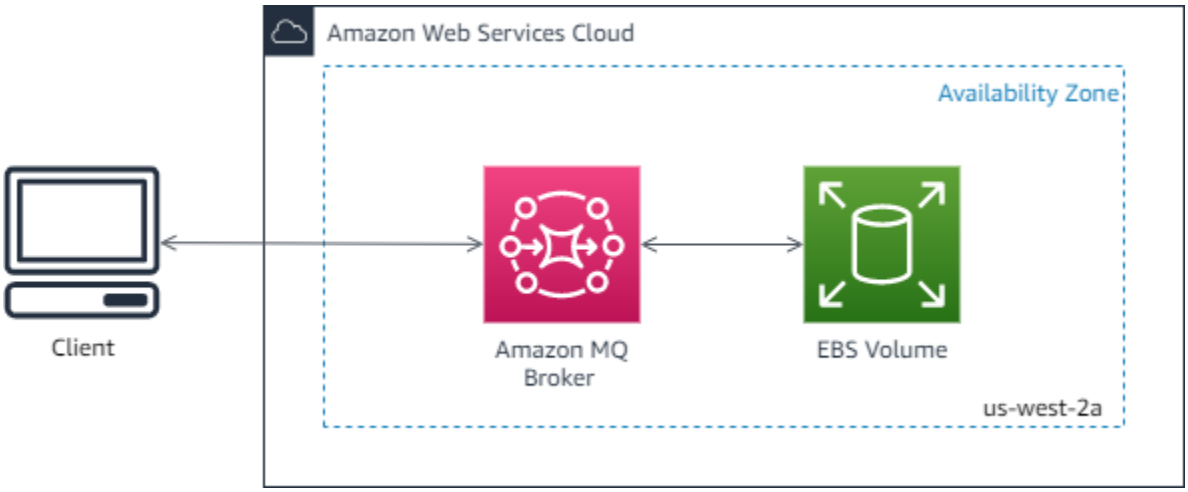


Diagram berikut menggambarkan broker instans tunggal dengan penyimpanan Amazon EBS yang direplikasi di beberapa server dalam satu AZ.



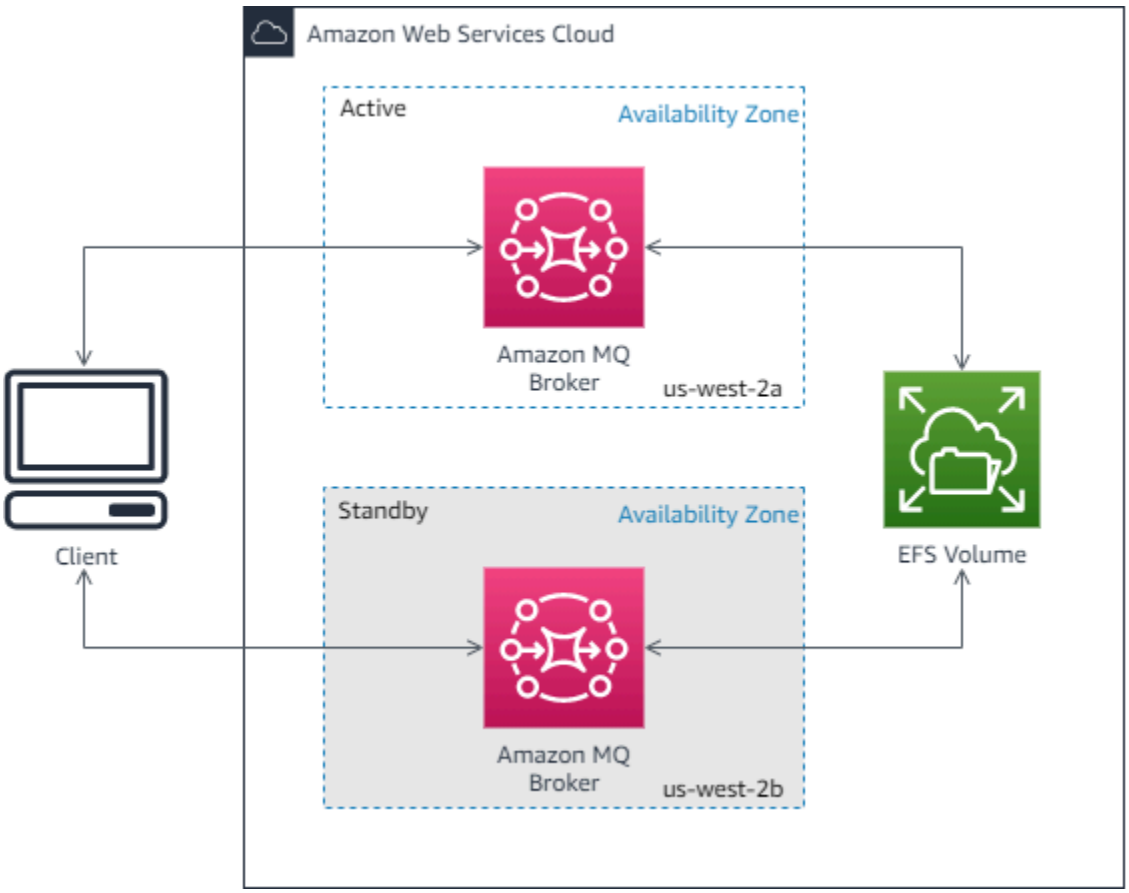
Opsi 2: active/standby Broker Amazon MQ untuk ketersediaan tinggi

Broker aktif/siaga terdiri dari dua broker di dua Availability Zone yang berbeda, dikonfigurasi dalam pasangan redundan. Broker ini berkomunikasi secara sinkron dengan aplikasi Anda dan Amazon EFS. Volume penyimpanan Amazon EFS dirancang untuk memberikan tingkat daya tahan dan ketersediaan tertinggi dengan menyimpan data secara berlebihan di beberapa Availability Zone ()AZs. Untuk informasi selengkapnya, lihat Storage.

Biasanya, hanya satu instans broker yang aktif setiap saat, sedangkan instans broker lainnya dalam status siaga. Jika salah satu instans broker malafungsi atau dalam pemeliharaan, dibutuhkan Amazon MQ beberapa saat untuk mengeluarkan instans tidak aktif dari layanan. Hal ini memungkinkan instans siaga sehat untuk menjadi aktif dan mulai menerima komunikasi masuk. Jendela pemeliharaan dan reboot broker yang Anda lakukan akan menyebabkan kegagalan terjadi. Ketika Anda mem-boot ulang broker, failover hanya berlangsung beberapa detik.

Untuk active/standby broker, Amazon MQ menyediakan dua ActiveMQ Web Console URLs, tetapi hanya satu URL yang aktif pada satu waktu. Demikian juga, Amazon MQ menyediakan dua titik akhir untuk setiap protokol tingkat wire, tetapi hanya satu titik akhir aktif di setiap pasangan pada satu waktu. Sufiks `-1` dan `-2` menunjukkan pasangan redundan. Untuk titik akhir protokol tingkat kabel, Anda harus mengizinkan aplikasi Anda terhubung ke salah satu titik akhir dengan menggunakan Failover Transport.

Diagram berikut menggambarkan active/standby broker dengan penyimpanan Amazon EFS direplikasi di beberapa. AZs



Jaringan broker Amazon MQ

Amazon MQ mendukung fitur jaringan broker ActiveMQ.

Jaringan broker terdiri dari beberapa broker atau broker instans tunggal yang aktif secara bersamaan. active/standby Membuat jaringan broker dapat meningkatkan ketersediaan, toleransi kesalahan, dan load balancing dengan beberapa contoh broker.

Bagaimana cara kerja Jaringan Pialang?

Jaringan broker didirikan dengan menghubungkan satu broker ke broker lain menggunakan konektor jaringan. Konektor jaringan menyediakan pesan sesuai permintaan dari satu broker ke broker lainnya. Konektor jaringan dikonfigurasi dalam konfigurasi broker sebagai koneksi non-dupleks atau dupleks. Untuk koneksi nondupleks, pesan diteruskan hanya dari satu broker ke broker lainnya. Untuk koneksi dupleks, pesan diteruskan dua arah antara kedua broker.

Jika konektor jaringan dikonfigurasi sebagai dupleks, pesan juga diteruskan dari Broker2 ke Broker1.

Anda dapat menggunakan koneksi non-dupleks dan dupleks dalam jaringan broker. Anda mungkin ingin memperkenalkan koneksi dupleks ke broker lain untuk meningkatkan lalu lintas, atau untuk menghindari peningkatan batas. Koneksi dupleks juga berguna untuk migrasi sebagian dari lokal ke broker terkelola Amazon MQ.

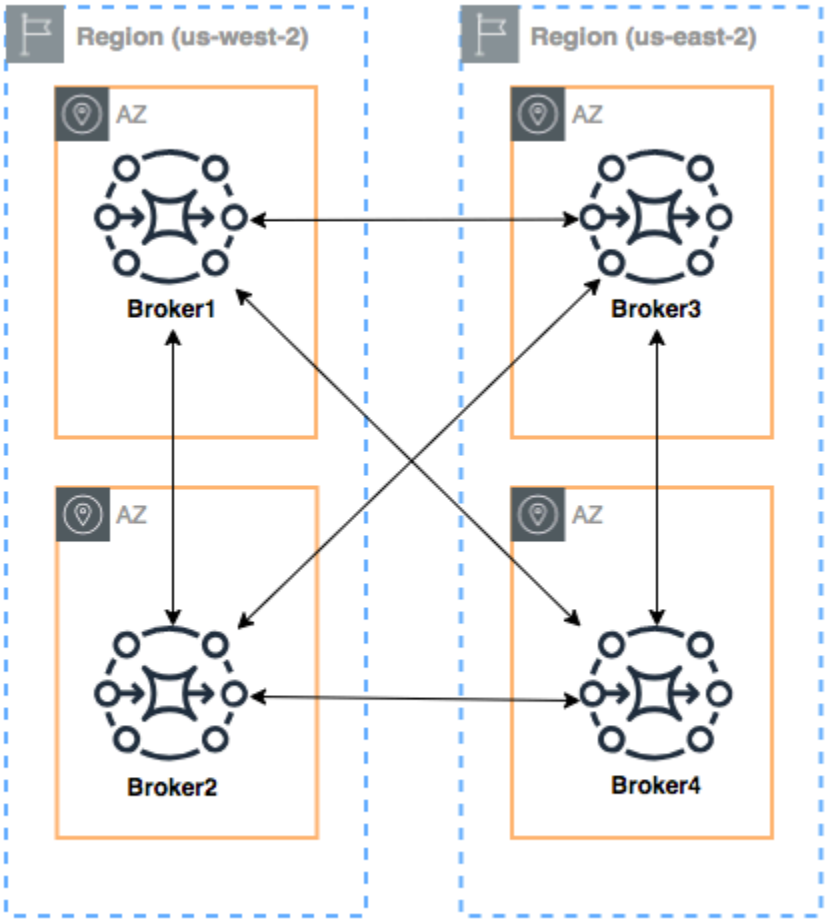
Bagaimana Cara Jaringan Broker Menangani Kredensial?

Agar broker A dapat terhubung ke broker B dalam jaringan, broker A harus menggunakan kredensial yang valid, seperti produsen atau konsumen lainnya. Alih-alih memberikan password di konfigurasi `<networkConnector>`broker A, Anda harus terlebih dahulu membuat pengguna di broker A dengan nilai yang sama seperti pengguna lain di broker B (ini adalah penguna terpisah yang unik serta berbagi nilai nama pengguna dan kata sandi yang sama). Saat Anda menentukan atribut `userName` dalam konfigurasi `<networkConnector>`, Amazon MQ akan menambahkan kata sandi secara otomatis pada saat waktu aktif.

> **Important**
>
> Jangan tentukan atribut `password` untuk `<networkConnector>`. Kami tidak merekomendasikan menyimpan kata sandi plaintext dalam file konfigurasi broker, karena ini membuat kata sandi terlihat di konsol Amazon MQ. Untuk informasi selengkapnya, lihat Configure Network Connectors for Your Broker.

Lintas wilayah

Untuk mengonfigurasi jaringan broker yang mencakup AWS wilayah, gunakan broker di wilayah tersebut, dan konfigurasikan konektor jaringan ke titik akhir broker tersebut.



Untuk mengonfigurasi jaringan broker seperti contoh ini, Anda dapat menambahkan entri `networkConnectors` ke konfigurasi Broker1 dan Broker4 yang mereferensikan titik akhir tingkat wire dari broker tersebut.

Konektor jaringan untuk Broker1:

```
<networkConnectors>
    <networkConnector name="1_to_2" userName="myCommonUser" duplex="true"
        uri="static:(ssl://b-9876l5k4-32ji-109h-8gfe-7d65c4b132a1-2.mq.us-
west-2.amazonaws.com:61617)"/>
    <networkConnector name="1_to_3" userName="myCommonUser" duplex="true"
        uri="static:(ssl://b-743c885d-2244-4c95-af67-a85017ff234e-3.mq.us-
east-2.amazonaws.com:61617)"/>
    <networkConnector name="1_to_4" userName="myCommonUser" duplex="true"
        uri="static:(ssl://b-62a7fb31-d51c-466a-a873-905cd660b553-4.mq.us-
east-2.amazonaws.com:61617)"/>
```

```
</networkConnectors>
```

Konektor jaringan untuk Broker2:

```
<networkConnectors>
    <networkConnector name="2_to_3" userName="myCommonUser" duplex="true"
        uri="static:(ssl://b-743c885d-2244-4c95-af67-a85017ff234e-3.mq.us-
east-2.amazonaws.com:61617)"/>
</networkConnectors>
```

Konektor jaringan untuk Broker4:

```
<networkConnectors>
    <networkConnector name="4_to_3" userName="myCommonUser" duplex="true"
        uri="static:(ssl://b-743c885d-2244-4c95-af67-a85017ff234e-3.mq.us-
east-2.amazonaws.com:61617)"/>
    <networkConnector name="4_to_2" userName="myCommonUser" duplex="true"
        uri="static:(ssl://b-9876l5k4-32ji-109h-8gfe-7d65c4b132a1-2.mq.us-
west-2.amazonaws.com:61617)"/>
</networkConnectors>
```

Failover Dinamis dengan Konektor Transportasi

Selain mengonfigurasi elemen `networkConnector`, Anda dapat mengonfigurasi opsi `transportConnector` broker untuk mengaktifkan failover dinamis, dan untuk menyeimbangkan kembali koneksi ketika broker ditambahkan atau dihapus dari jaringan.

```
<transportConnectors>
  <transportConnector name="openwire" updateClusterClients="true"
 rebalanceClusterClients="true" updateClusterClientsOnRemove="true"/>
</transportConnectors>
```

Dalam contoh ini, `updateClusterClients` dan `rebalanceClusterClients` diatur ke `true`. Di sini, klien akan diberikan daftar broker dalam jaringan, dan akan meminta mereka untuk menyeimbangkan kembali jika broker baru bergabung.

Opsi yang tersedia:

- `updateClusterClients`: Meneruskan informasi ke klien tentang perubahan dalam topologi jaringan broker.

- `rebalanceClusterClients`: Membuat klien menyeimbangkan ulang di seluruh broker ketika broker baru ditambahkan ke jaringan broker.
- `updateClusterClientsOnRemove`: Memberi klien informasi topologi terbaru ketika broker meninggalkan jaringan broker.

Saat `updateClusterClients` diatur ke True, klien dapat dikonfigurasi untuk terhubung ke broker tunggal dalam jaringan broker.

```
failover:(ssl://b-1234a5b6-78cd-901e-2fgh-3i45j6k178l9-1.mq.us-
east-2.amazonaws.com:61617)
```

Ketika broker baru terhubung, ia akan menerima URIs daftar semua broker di jaringan. Jika koneksi ke broker gagal, secara dinamis broker dapat beralih ke salah satu broker yang disediakan ketika terhubung.

Untuk informasi selengkapnya tentang failover, lihat Opsi Sisi Broker untuk Failover dalam dokumentasi ActiveMQ.

Amazon MQ untuk jenis instans broker ActiveMQ

Deskripsi gabungan dari kelas instance broker (`m5`) dan size (`large`,`medium`) disebut tipe instance broker (misalnya,`mq.m5.large`). Tabel berikut mencantumkan jenis instans broker Amazon MQ yang tersedia untuk broker ActiveMQ.

Amazon MQ menyediakan setidaknya pemberitahuan 90 hari sebelum jenis instans mencapai akhir dukungan. Kami merekomendasikan untuk meningkatkan broker Anda ke jenis instans baru sebelum end-of-support tanggal untuk mencegah gangguan apa pun.

> **Important**
>
> Anda tidak dapat membuat broker pada `t2.micro` atau `mq.m4.large` setelah 17 Maret 2025.

Tipe Instans	vCPU	Memori (GiB)	Penggunaan yang Direkomendasikan	Penyimpanan	Akhir dukungan di Amazon MQ
`mq.t3.micro`	2	1	Evaluasi	EFS	
`mq.m5.large`	2	8	Produksi	EFS atau EBS	
`mq.m5.xlarge`	4	16	Produksi	EFS atau EBS	
`mq.m5.2xlarge`	8	32	Produksi	EFS atau EBS	
`mq.m5.4xlarge`	16	64	Produksi	EFS atau EBS	

Untuk informasi selengkapnya tentang pertimbangan throughput, lihat Memilih Tipe Instans Broker yang Tepat untuk Throughput Terbaik.

Amazon MQ untuk konfigurasi broker ActiveMQ

Konfigurasi berisi semua pengaturan untuk broker ActiveMQ Anda dalam format XHTML (mirip dengan file ActiveMQ). `activemq.xml` Anda dapat membuat konfigurasi sebelum membuat broker. Kemudian Anda dapat menerapkan konfigurasi ke satu atau lebih broker.

> **Important**
>
> Pembuatan perubahan pada konfigurasi tidak akan segera menerapkan perubahan ke broker. Untuk menerapkan perubahan Anda, Anda harus menunggu jendela pemeliharaan berikutnya atau reboot broker.
>
> Anda hanya dapat menghapus konfigurasi menggunakan `DeleteConfiguration` API. Untuk informasi selengkapnya, lihat Konfigurasi di Referensi API Amazon MQ.

Atribut

Konfigurasi broker memiliki beberapa atribut, misalnya:

- Nama (`MyConfiguration`)
- ID (`c-1234a5b6-78cd-901e-2fgh-3i45j6k178l9`)
- Amazon Resource Name (ARN) (`arn:aws:mq:us-east-2:123456789012:configuration:c-1234a5b6-78cd-901e-2fgh-3i45j6k178l9`)

Untuk daftar lengkap atribut konfigurasi, lihat di Referensi REST API Amazon MQ:

- ID Operasi REST: Konfigurasi
- ID Operasi REST: Konfigurasi

Untuk daftar lengkap atribut revisi konfigurasi, lihat hal berikut:

- ID Operasi REST: Revisi Konfigurasi
- ID Operasi REST: Revisi Konfigurasi

Menggunakan file konfigurasi Spring XML

Broker ActiveMQ dikonfigurasi menggunakan file Spring XML. Anda dapat mengonfigurasi berbagai aspek broker ActiveMQ, seperti tujuan yang telah ditetapkan, kebijakan tujuan, kebijakan otorisasi, dan plugin. Amazon MQ mengontrol beberapa elemen konfigurasi tersebut, seperti transportasi jaringan dan penyimpanan. Opsi konfigurasi lainnya, seperti membuat jaringan broker, saat ini tidak didukung.

Kumpulan lengkap opsi konfigurasi yang didukung ditentukan dalam skema XML Amazon MQ. Unduh file zip dari skema yang didukung menggunakan tautan berikut.

- `amazon-mq-active-mq-5.19.1.xsd.zip`
- `amazon-mq-active-mq-5.18.4.xsd.zip`
- `amazon-mq-active-mq-5.17.6.xsd.zip`
- `amazon-mq-active-mq-5.16.7.xsd.zip`
- `amazon-mq-active-mq-5.15.16.xsd.zip`

Anda dapat menggunakan skema ini untuk memvalidasi dan membersihkan file konfigurasi. Amazon MQ juga memungkinkan Anda menyediakan konfigurasi dengan mengunggah file XML. Ketika Anda mengunggah file XML, Amazon MQ secara otomatis membersihkan serta menghapus parameter konfigurasi yang tidak valid dan dilarang sesuai dengan skema.

> **Note**
>
> Anda hanya dapat menggunakan nilai statis untuk atribut. Amazon MQ membersihkan elemen dan atribut yang berisi ekspresi Spring, variabel, serta referensi elemen dari konfigurasi Anda.

Membuat Amazon MQ untuk konfigurasi broker ActiveMQ

Konfigurasi berisi semua pengaturan untuk broker ActiveMQ Anda, dalam format XML (mirip dengan ActiveMQ file `activemq.xml`). Anda dapat membuat konfigurasi sebelum membuat broker. Kemudian Anda dapat menerapkan konfigurasi ke satu atau lebih broker. Anda dapat segera menerapkan konfigurasi atau selama jendela pemeliharaan.

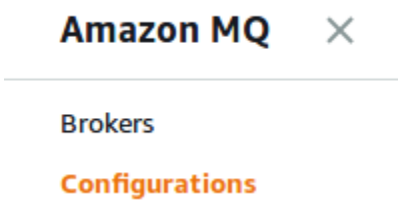
Contoh berikut menunjukkan cara membuat dan menerapkan konfigurasi broker Amazon MQ menggunakan Konsol Manajemen AWS.

> **Important**
>
> Anda hanya dapat menghapus konfigurasi menggunakan `DeleteConfiguration` API. Untuk informasi selengkapnya, lihat Konfigurasi di Referensi API Amazon MQ.

Buat Konfigurasi Baru

Untuk membuat konfigurasi broker baru, pertama buat konfigurasi baru.

1. Masuk ke konsol Amazon MQ.
2. Di sebelah kiri, perluas panel navigasi dan pilih Konfigurasi.



3. Di halaman Konfigurasi, pilih Buat konfigurasi.
4. Di halaman Buat konfigurasi, pada bagian Detail, ketik Nama konfigurasi (Misalnya, `MyConfiguration`) dan pilih versi Mesin broker.

> **Note**
>
> Untuk mempelajari lebih lanjut tentang versi mesin ActiveMQ yang didukung oleh Amazon MQ untuk ActiveMQ, lihat. the section called “Manajemen versi”

5. Pilih Buat konfigurasi.

Buat Revisi Konfigurasi Baru

Setelah Anda membuat konfigurasi broker, Anda perlu mengedit konfigurasi menggunakan revisi konfigurasi.

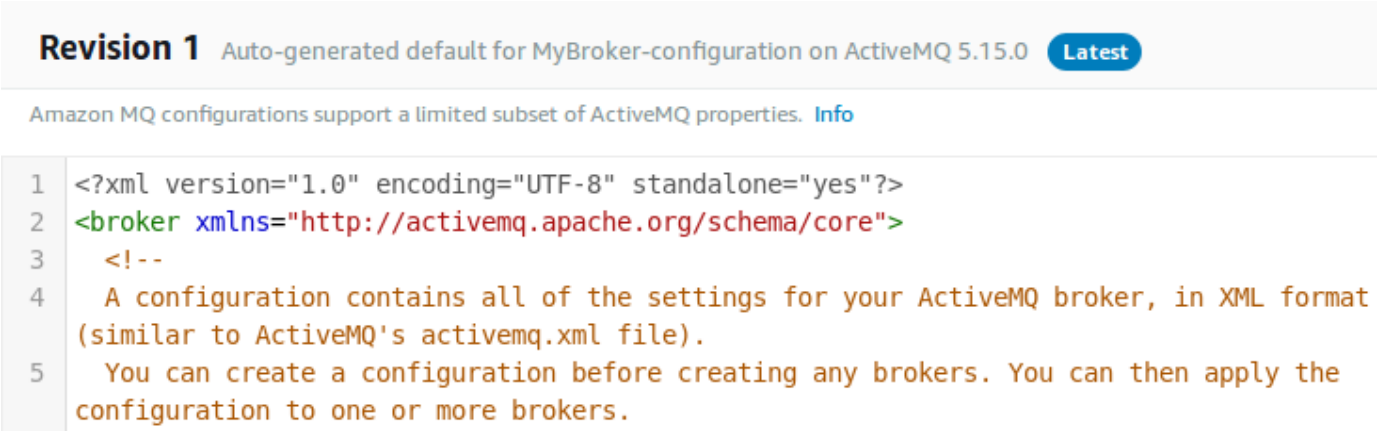
1. Dari daftar konfigurasi, pilih ***`MyConfiguration`***.

> **Note**
>
> Revisi konfigurasi pertama selalu dibuat untuk Anda ketika Amazon MQ membuat konfigurasi.

Pada ***`MyConfiguration`***halaman, jenis dan versi mesin broker yang digunakan revisi konfigurasi baru Anda (misalnya, Apache ActiveMQ 5.15.16) ditampilkan.

2. Di tab Detail konfigurasi, nomor revisi konfigurasi, deskripsi, dan konfigurasi broker dalam format XML akan ditampilkan.

> **Note**
>
> Mengedit konfigurasi saat ini membuat revisi konfigurasi baru.



3. Pilih Edit konfigurasi dan buat perubahan pada konfigurasi XML.
4. Pilih Simpan.

   Kotak dialog Simpan revisi akan ditampilkan.
5. (Opsional) Tipe `A description of the changes in this revision`.
6. Pilih Simpan.

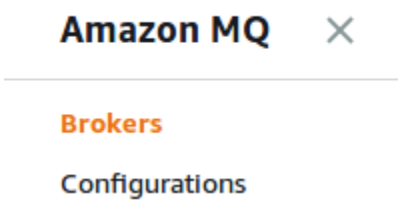
   Revisi konfigurasi baru akan disimpan.

> **Important**
>
> Konsol Amazon MQ secara otomatis membersihkan parameter konfigurasi yang tidak valid dan dilarang sesuai dengan skema. Untuk informasi selengkapnya dan daftar lengkap parameter XML yang diizinkan, lihat Amazon MQ Broker Configuration Parameters.

Terapkan Revisi Konfigurasi ke Broker Anda

Setelah merevisi konfigurasi, Anda dapat menerapkan revisi konfigurasi ke broker Anda.

1. Di sebelah kiri, perluas panel navigasi dan pilih Broker.



2. Dari daftar broker, pilih broker Anda (misalnya, MyBroker) dan kemudian pilih Edit.
3. Pada *MyBroker* halaman Edit, di bagian Konfigurasi, pilih Konfigurasi dan Revisi dan kemudian pilih Jadwal Modifikasi.
4. Di bagian Jadwalkan perubahan broker, pilih apakah akan menerapkan perubahan Selama jendela pemeliharaan terjadwal berikutnya atau Segera.

> **Important**
>
> Pialang instans tunggal sedang offline saat di-boot ulang. Untuk broker cluster, hanya satu node yang turun pada satu waktu sementara broker melakukan reboot.

5. Pilih Terapkan.

   Revisi konfigurasi Anda diterapkan ke broker pada waktu yang ditentukan.

Edit Amazon MQ untuk revisi konfigurasi ActiveMQ

Anda mungkin ingin mengedit revisi konfigurasi setelah menerapkannya ke broker Anda. Gunakan petunjuk berikut untuk mengedit revisi konfigurasi.

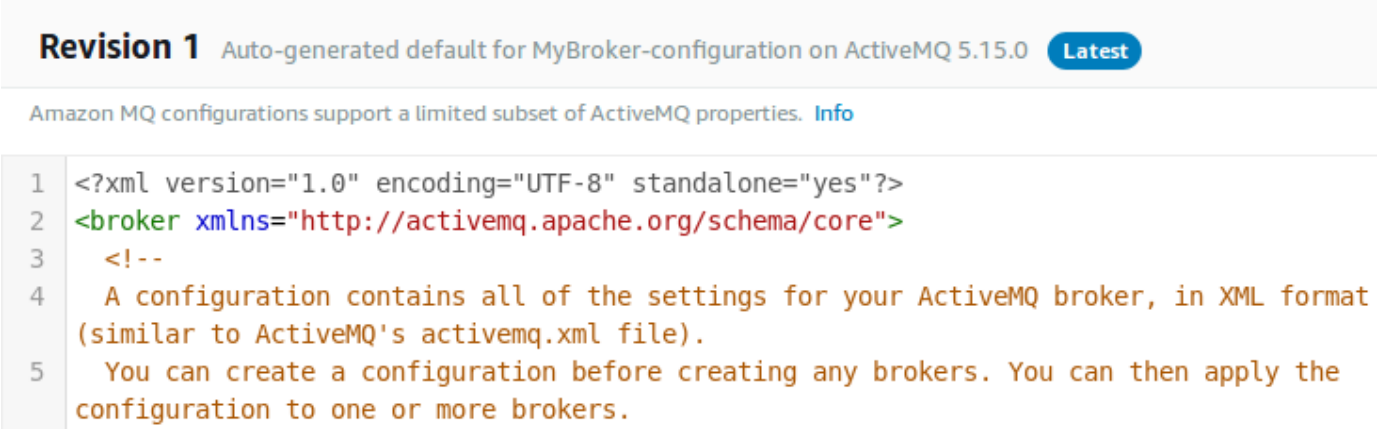
1. Masuk ke konsol Amazon MQ.
2. Dari daftar broker, pilih broker Anda (misalnya, MyBroker) dan kemudian pilih Edit.
3. Pada ***MyBroker***halaman, pilih Edit.
4. Pada *MyBroker* halaman Edit, di bagian Konfigurasi, pilih Konfigurasi dan Revisi dan kemudian pilih Edit.

> **Note**
>
> Kecuali Anda memilih konfigurasi ketika membuat broker, revisi konfigurasi pertama selalu dibuat untuk Anda ketika Amazon MQ membuat broker.

Pada ***MyBroker***halaman, jenis dan versi mesin broker yang digunakan konfigurasi (misalnya, Apache ActiveMQ 5.15.8) ditampilkan.

5. Di tab Detail konfigurasi, nomor revisi konfigurasi, deskripsi, dan konfigurasi broker dalam format XML akan ditampilkan.

> **Note**
>
> Mengedit konfigurasi saat ini membuat revisi konfigurasi baru.



6. Pilih Edit konfigurasi dan buat perubahan pada konfigurasi XML.
7. Pilih Simpan.

   Kotak dialog Simpan revisi akan ditampilkan.
8. (Opsional) Tipe `A description of the changes in this revision`.
9. Pilih Simpan.

   Revisi konfigurasi baru akan disimpan.

> **Important**
>
> Konsol Amazon MQ secara otomatis membersihkan parameter konfigurasi yang tidak valid dan dilarang sesuai dengan skema. Untuk informasi selengkapnya dan daftar lengkap parameter XML yang diizinkan, lihat Amazon MQ Broker Configuration Parameters.

Elemen diizinkan dalam konfigurasi Amazon MQ

Berikut adalah daftar detail dari elemen yang diizinkan dalam konfigurasi Amazon MQ. Untuk informasi selengkapnya, lihat Konfigurasi XML dalam dokumentasi Apache ActiveMQ.

Elemen
`abortSlowAckConsumerStrategy` (atribut)
`abortSlowConsumerStrategy` (atribut)
`authorizationEntry` (atribut)
`authorizationMap` (elemen pengumpulan anak)
`authorizationPlugin` (elemen pengumpulan anak)
`broker` (atribut | elemen pengumpulan anak)
`cachedMessageGroupMapFactory` (atribut)
`compositeQueue` (atribut | elemen pengumpulan anak)
`compositeTopic` (atribut | elemen pengumpulan anak)
`constantPendingMessageLimitStrategy` (atribut)
`discarding` (atribut)
`discardingDLQBrokerPlugin` (atribut)
`fileCursor`
`fileDurableSubscriberCursor`
`fileQueueCursor`
`filteredDestination` (atribut)
`fixedCountSubscriptionRecoveryPolicy` (atribut)

Elemen
`fixedSizedSubscriptionRecoveryPolicy` (atribut)
`forcePersistencyModeBrokerPlugin` (atribut)
`individualDeadLetterStrategy` (atribut)
`lastImageSubscriptionRecoveryPolicy`
`messageGroupHashBucketFactory` (atribut)
`mirroredQueue` (atribut)
`noSubscriptionRecoveryPolicy`
`oldestMessageEvictionStrategy` (atribut)
`oldestMessageWithLowestPriorityEvictionStrategy` (atribut)
`policyEntry` (atribut | elemen pengumpulan anak)
`policyMap` (elemen pengumpulan anak)
`prefetchRatePendingMessageLimitStrategy` (atribut)
`priorityDispatchPolicy`
`priorityNetworkDispatchPolicy`
`queryBasedSubscriptionRecoveryPolicy` (atribut)
`queue` (atribut)
`redeliveryPlugin` (atribut | elemen pengumpulan anak)
`redeliveryPolicy` (atribut)
`redeliveryPolicyMap` (elemen pengumpulan anak)
`retainedMessageSubscriptionRecoveryPolicy` (elemen pengumpulan anak)

Elemen
roundRobinDispatchPolicy
sharedDeadLetterStrategy (atribut | elemen pengumpulan anak)
simpleDispatchPolicy
simpleMessageGroupMapFactory
statisticsBrokerPlugin
storeCursor
storeDurableSubscriberCursor (atribut)
strictOrderDispatchPolicy
tempDestinationAuthorizationEntry (atribut)
tempQueue (atribut)
tempTopic (atribut)
timedSubscriptionRecoveryPolicy (atribut)
timeStampingBrokerPlugin (atribut)
topic (atribut)
transportConnector (atribut)
uniquePropertyMessageEvictionStrategy (atribut)
virtualDestinationInterceptor (elemen pengumpulan anak)
virtualTopic (atribut)
vmCursor
vmDurableCursor

Elemen
vmQueueCursor

Elemen dan Atribut yang Diizinkan dalam Konfigurasi Amazon MQ

Berikut adalah daftar detail dari elemen dan atribut yang diizinkan dalam konfigurasi Amazon MQ. Untuk informasi selengkapnya, lihat Konfigurasi XML dalam dokumentasi Apache ActiveMQ.

Elemen	Atribut
abortSlowAckConsumerStrategy	abortConnection
	checkPeriod
	ignoreIdleConsumers
	ignoreNetworkConsumers
	maxSlowCount
	maxSlowDuration
	maxTimeSinceLastAck
	name
abortSlowConsumerStrategy	abortConnection
	checkPeriod
	ignoreNetworkConsumers
	maxSlowCount
	maxSlowDuration
	name
authorizationEntry	admin

Elemen	Atribut
	queue
	read
	tempQueue
	tempTopic
	topic
	write
broker	advisorySupport
	allowTempAutoCreationOnSend
	cacheTempDestinations
	consumerSystemUsagePortion
	dedicatedTaskRunner
	deleteAllMessagesOnStartup
	keepDurableSubsActive
	enableMessageExpirationOnActiveDurableSubs
	maxPurgedDestinationsPerSweep
	maxSchedulerRepeatAllowed
	monitorConnectionSplits
	networkConnectorStartAsync
	offlineDurableSubscriberTaskSchedule

Elemen	Atribut
	offlineDurableSubscriberTimeout
	persistenceThreadPriority
	persistent
	populateJMSXUserID
	producerSystemUsagePortion
	rejectDurableConsumers
	rollbackOnlyOnAsyncException
	schedulePeriodForDestinatio nPurge
	schedulerSupport
	splitSystemUsageForProducer sConsumers
	taskRunnerPriority
	timeBeforePurgeTempDestinations
	useAuthenticatedPrincipalFo rJMSXUserID
	useMirroredQueues
	useTempMirroredQueues
	useVirtualDestSubs
	useVirtualDestSubsOnCreation
	useVirtualTopics

Elemen	Atribut
`cachedMessageGroupMapFactory`	`cacheSize`
`compositeQueue`	`concurrentSend`
	`copyMessage`
	`forwardOnly`
	`name`
	`sendWhenNotMatched`
`compositeTopic`	`concurrentSend`
	`copyMessage`
	`forwardOnly`
	`name`
	`sendWhenNotMatched`
conditionalNetworkBridgeFilterFactory	`rateDuration`
	`rateLimit`
	`replayDelay`
	`replayWhenNoConsumers`
	`selectorAware` Didukung di Apache ActiveMQ 5.16.x
`constantPendingMessageLimitStrategy`	`limit`

Elemen	Atribut
discarding	deadLetterQueue
	enableAudit
	expiration
	maxAuditDepth
	maxProducersToAudit
	processExpired
	processNonPersistent
discardingDLQBrokerPlugin	dropAll
	dropOnly
	dropTemporaryQueues
	dropTemporaryTopics
	reportInterval
filteredDestination	queue
	selector
	topic
fixedCountSubscriptionRecoveryPolicy	maximumSize
fixedSizedSubscriptionRecoveryPolicy	maximumSize
	useSharedBuffer
forcePersistencyModeBrokerPlugin	persistenceFlag
individualDeadLetterStrategy	destinationPerDurableSubscriber

Elemen	Atribut
	enableAudit
	expiration
	maxAuditDepth
	maxProducersToAudit
	processExpired
	processNonPersistent
	queuePrefix
	queueSuffix
	topicPrefix
	topicSuffix
	useQueueForQueueMessages
	useQueueForTopicMessages
messageGroupHashBucketFactory	bucketCount
	cacheSize
mirroredQueue	copyMessage
	postfix
	prefix
oldestMessageEvictionStrategy	evictExpiredMessagesHighWat ermark
oldestMessageWithLowestPrio rityEvictionStrategy	evictExpiredMessagesHighWat ermark

Elemen	Atribut
policyEntry	advisoryForConsumed
	advisoryForDelivery
	advisoryForDiscardingMessages
	advisoryForFastProducers
	advisoryForSlowConsumers
	advisoryWhenFull
	allConsumersExclusiveByDefault
	alwaysRetroactive
	blockedProducerWarningInterval
	consumersBeforeDispatchStarts
	cursorMemoryHighWaterMark
	doOptimzeMessageStorage
	durableTopicPrefetch
	enableAudit
	expireMessagesPeriod
	gcInactiveDestinations
	gcWithNetworkConsumers
	inactiveTimeoutBeforeGC
	inactiveTimoutBeforeGC
	includeBodyForAdvisory

Elemen	Atribut
	lazyDispatch
	maxAuditDepth
	maxBrowsePageSize
	maxDestinations
	maxExpirePageSize
	maxPageSize
	maxProducersToAudit
	maxQueueAuditDepth
	memoryLimit
	messageGroupMapFactoryType
	minimumMessageSize
	optimizedDispatch
	optimizeMessageStoreInFligh tLimit
	persistJMSRedelivered
	prioritizedMessages
	producerFlowControl
	queue
	queueBrowserPrefetch
	queuePrefetch
	reduceMemoryFootprint

Elemen	Atribut
	`sendAdvisoryIfNoConsumers`
	`sendFailIfNoSpace`
	`sendFailIfNoSpaceAfterTimeout` Didukung di Apache ActiveMQ 5.16.4 dan di atas
	`sendDuplicateFromStoreToDLQ`
	`storeUsageHighWaterMark`
	`strictOrderDispatch`
	`tempQueue`
	`tempTopic`
	`timeBeforeDispatchStarts`
	`topic`
	`topicPrefetch`
	`useCache`
	`useConsumerPriority`
`usePrefetchExtension`	
`prefetchRatePendingMessageLimitStrategy`	`multiplier`
`queryBasedSubscriptionRecoveryPolicy`	`query`

Elemen	Atribut
queue	DLQ
	physicalName
redeliveryPlugin	fallbackToDeadLetter
	sendToDlqIfMaxRetriesExceeded
redeliveryPolicy	backOffMultiplier
	collisionAvoidancePercent
	initialRedeliveryDelay
	maximumRedeliveries
	maximumRedeliveryDelay
	preDispatchCheck
	queue
	redeliveryDelay
	tempQueue
	tempTopic
	topic
	useCollisionAvoidance
	useExponentialBackOff
sharedDeadLetterStrategy	enableAudit
	expiration
	maxAuditDepth

Elemen	Atribut
storeDurableSubscriberCursor	maxProducersToAudit
	processExpired
	processNonPersistent
	immediatePriorityDispatch
	useCache
tempDestinationAuthorizationEntry	admin
	queue
	read
	tempQueue
	tempTopic
	topic
	write
tempQueue	DLQ
	physicalName
tempTopic	DLQ
	physicalName
timedSubscriptionRecoveryPolicy	zeroExpirationOverride
timeStampingBrokerPlugin	recoverDuration
	futureOnly
	processNetworkMessages

Elemen	Atribut
	ttlCeiling
topic	DLQ
	physicalName
transportConnector	name
	updateClusterClients
	rebalanceClusterClients
	updateClusterClientsOnRemove
uniquePropertyMessageEvictionStrategy	evictExpiredMessagesHighWatermark
	propertyName
virtualTopic	concurrentSend
	local
	dropOnResourceLimit
	name
	postfix
	prefix
	selectorAware
	setOriginalDestination
	transactedSend

Atribut Elemen Induk Amazon MQ

Berikut adalah penjelasan detail tentang atribut elemen induk. Untuk informasi selengkapnya, lihat Konfigurasi XML dalam dokumentasi Apache ActiveMQ.

Topik

- makelar

makelar

`broker` adalah elemen pengumpulan induk.

Atribut

networkConnectionStartAsinkron

Untuk memitigasi latensi jaringan dan memungkinkan jaringan lain memulai secara tepat waktu, gunakan tanda `<networkConnectionStartAsync>`. Tanda menginstruksikan broker untuk menggunakan eksekutor guna memulai koneksi jaringan secara paralel, asinkron dengan memulai broker.

Default: `false`

Contoh Konfigurasi

```
<broker networkConnectorStartAsync="false"/>
```

Elemen, Elemen Pengumpulan Anak, dan Elemen Anaknya yang Diizinkan dalam Konfigurasi Amazon MQ

Berikut adalah daftar detail elemen, elemen pengumpulan anak, dan elemen anaknya yang diizinkan dalam konfigurasi Amazon MQ. Untuk informasi selengkapnya, lihat Konfigurasi XML dalam dokumentasi Apache ActiveMQ.

Elemen	Elemen Pengumpulan Anak	Elemen Anak
`authorizationMap`	`authorizationEntries`	`authorizationEntry`
		`tempDestinationAuthorizationEntry`

Elemen	Elemen Pengumpulan Anak	Elemen Anak
	defaultEntry	authorizationEntry
		tempDestinationAuthorizationEntry
	tempDestinationAuthorizationEntry	tempDestinationAuthorizationEntry
authorizationPlugin	map	authorizationMap
broker	destinationInterceptors	mirroredQueue
		virtualDestinationInterceptor
	destinationPolicy	policyMap
	destinations	queue
		tempQueue
		tempTopic
		topic
	networkConnectors	networkConnector
	persistenceAdapter	kahaDB
	plugins	authorizationPlugin
		discardingDLQBrokerPlugin
		forcePersistencyModeBrokerPlugin
		redeliveryPlugin

Elemen	Elemen Pengumpulan Anak	Elemen Anak
		statisticsBrokerPlugin
		timeStampingBrokerPlugin
	systemUsage	systemUsage
	transportConnector	name
		updateClusterClients
		rebalanceClusterClients
		updateClusterClientsOnRemove
compositeQueue	forwardTo	queue
		tempQueue
		tempTopic
		topic
		filteredDestination
compositeTopic	forwardTo	queue
		tempQueue
		tempTopic
		topic
		filteredDestination
policyEntry	deadLetterStrategy	discarding

Elemen	Elemen Pengumpulan Anak	Elemen Anak
		individualDeadLetterStrategy
		sharedDeadLetterStrategy
	destination	queue
		tempQueue
		tempTopic
		topic
	dispatchPolicy	priorityDispatchPolicy
		priorityNetworkDispatchPolicy
		roundRobinDispatchPolicy
		simpleDispatchPolicy
		strictOrderDispatchPolicy
		clientIdFilterDispatchPolicy
	messageEvictionStrategy	oldestMessageEvictionStrategy
		oldestMessageWithLowestPriorityEvictionStrategy

Elemen	Elemen Pengumpulan Anak	Elemen Anak
		uniquePropertyMessageEvictionStrategy
	messageGroupMapFactory	cachedMessageGroupMapFactory
		messageGroupHashBucketFactory
		simpleMessageGroupMapFactory
	pendingDurableSubscriberPolicy	fileDurableSubscriberCursor
		storeDurableSubscriberCursor
		vmDurableCursor
	pendingMessageLimitStrategy	constantPendingMessageLimitStrategy
		prefetchRatePendingMessageLimitStrategy
	pendingQueuePolicy	fileQueueCursor
		storeCursor
		vmQueueCursor
	pendingSubscriberPolicy	fileCursor
		vmCursor

Elemen	Elemen Pengumpulan Anak	Elemen Anak
	slowConsumerStrategy	abortSlowAckConsumerStrategy
		abortSlowConsumerStrategy
	subscriptionRecoveryPolicy	fixedCountSubscriptionRecoveryPolicy
		fixedSizedSubscriptionRecoveryPolicy
		lastImageSubscriptionRecoveryPolicy
		noSubscriptionRecoveryPolicy
		queryBasedSubscriptionRecoveryPolicy
		retainedMessageSubscriptionRecoveryPolicy
timedSubscriptionRecoveryPolicy		
policyMap	defaultEntry	policyEntry
	policyEntries	policyEntry
redeliveryPlugin	redeliveryPolicyMap	redeliveryPolicyMap
redeliveryPolicyMap	defaultEntry	redeliveryPolicy
	redeliveryPolicyEntries	redeliveryPolicy

Elemen	Elemen Pengumpulan Anak	Elemen Anak
retainedMessageSubscriptionRecoveryPolicy	wrapped	fixedCountSubscriptionRecoveryPolicy
		fixedSizedSubscriptionRecoveryPolicy
		lastImageSubscriptionRecoveryPolicy
		noSubscriptionRecoveryPolicy
		queryBasedSubscriptionRecoveryPolicy
		retainedMessageSubscriptionRecoveryPolicy
		timedSubscriptionRecoveryPolicy
sharedDeadLetterStrategy	deadLetterQueue	queue
		tempQueue
		tempTopic
		topic
virtualDestinationInterceptor	virtualDestinations	compositeQueue
		compositeTopic
		virtualTopic

Atribut Elemen Anak Amazon MQ

Berikut adalah penjelasan detail tentang atribut elemen anak. Untuk informasi selengkapnya, lihat Konfigurasi XML dalam dokumentasi Apache ActiveMQ.

Topik

- authorizationEntry
- networkConnector
- kahaDB
- systemUsage

authorizationEntry

`authorizationEntry` adalah anak dari elemen pengumpulan anak `authorizationEntries`.

Atribut

admin|baca|tulis

Izin yang diberikan kepada grup pengguna. Untuk informasi selengkapnya, lihat Selalu konfigurasikan peta otorisasi.

Jika menentukan peta otorisasi yang tidak menyertakan grup `activemq-webconsole`, Anda tidak dapat menggunakan Konsol Web ActiveMQ karena grup tidak berwenang untuk mengirim pesan ke, atau menerima pesan dari, broker Amazon MQ.

Default: `null`

Contoh Konfigurasi

```
<authorizationPlugin>
                                <map>
                                    <authorizationMap>
                                        <authorizationEntries>
                                            <authorizationEntry admin="admins,activemq-
webconsole" read="admins,users,activemq-webconsole" write="admins,activemq-webconsole"
 queue=">"/>
                                            <authorizationEntry admin="admins,activemq-
webconsole" read="admins,users,activemq-webconsole" write="admins,activemq-webconsole"
 topic=">"/>
```

```
                                    </authorizationEntries>
                                </authorizationMap>
                            </map>
                            </authorizationPlugin>
```

> **Note**
>
> `activemq-webconsole`Grup di ActiveMQ di Amazon MQ memiliki izin admin pada semua antrian dan topik. Semua pengguna dalam grup ini akan memiliki akses admin.

networkConnector

`networkConnector` adalah anak dari elemen pengumpulan anak `networkConnectors`.

Topik

- Atribut
- Contoh Konfigurasi

Atribut

conduitSubscriptions

Menentukan apakah koneksi jaringan dalam jaringan broker memperlakukan sejumlah konsumen yang berlangganan ke tujuan yang sama sebagai satu konsumen. Misalnya, jika `conduitSubscriptions` diatur ke `true` dan dua konsumen terhubung ke broker B dan mengonsumsi dari tujuan, broker B menggabungkan langganan ke langganan logis tunggal melalui koneksi jaringan ke broker A, sehingga hanya satu salinan pesan yang diteruskan dari broker A ke broker B.

> **Note**
>
> Mengatur `conduitSubscriptions` ke `true` dapat mengurangi lalu lintas jaringan redundan. Namun, menggunakan atribut ini dapat menimbulkan implikasi untuk penyeimbangan beban pesan di seluruh konsumen dan mungkin menyebabkan perilaku yang salah dalam skenario tertentu (misalnya, dengan penyeleksi pesan JMS atau dengan topik yang tahan lama).

Default: `true`

dupleks

Menentukan apakah koneksi dalam jaringan broker digunakan untuk memproduksi dan mengonsumsi pesan. Sebagai contoh, jika broker A membuat koneksi ke broker B dalam modus nondupleks, pesan dapat diteruskan hanya dari broker A ke broker B. Namun, jika broker A membuat koneksi dupleks ke broker B, broker B dapat meneruskan pesan ke broker A tanpa harus mengonfigurasi `<networkConnector>`.

Default: `false`

name

Nama jembatan dalam jaringan broker.

Default: `bridge`

uri

Titik akhir protokol tingkat wire untuk salah satu dari dua broker (atau untuk beberapa broker) dalam jaringan broker.

Default: `null`

nama pengguna

Nama pengguna umum untuk broker dalam jaringan broker.

Default: `null`

Contoh Konfigurasi

> **Note**
>
> Saat menggunakan `networkConnector` untuk menentukan jaringan broker, jangan sertakan kata sandi untuk pengguna yang umum bagi broker Anda.

Jaringan Broker dengan Dua Broker

Dalam konfigurasi ini, dua broker terhubung dalam jaringan broker. Nama konektor jaringan adalah`connector_1_to_2`, nama pengguna yang umum untuk broker adalah`myCommonUser`,

koneksi`duplex`, dan URI OpenWire titik akhir diawali oleh`static:`, menunjukkan one-to-one hubungan antara broker.

```
<networkConnectors>
                                    <networkConnector name="connector_1_to_2"
 userName="myCommonUser" duplex="true"
                                        uri="static:(ssl://
b-1234a5b6-78cd-901e-2fgh-3i45j6k178l9-1.mq.us-east-2.amazonaws.com:61617)"/>
                                    </networkConnectors>
```

Untuk informasi selengkapnya, lihat Configure Network Connectors for Your Broker.

Jaringan Broker dengan Beberapa Broker

Dalam konfigurasi ini, beberapa broker terhubung dalam jaringan broker. Nama konektor jaringan adalah`connector_1_to_2`, nama pengguna yang umum untuk broker adalah`myCommonUser`, koneksi`duplex`, dan daftar OpenWire titik akhir yang dipisahkan koma URIs diawali oleh`masterslave:`, menunjukkan koneksi failover antara broker. Failover dari broker ke broker tidak terjadi secara acak dan upaya koneksi ulang terus berlangsung tanpa batas.

```
<networkConnectors>
                                    <networkConnector name="connector_1_to_2"
 userName="myCommonUser" duplex="true"
                                        uri="masterslave:(ssl://
b-1234a5b6-78cd-901e-2fgh-3i45j6k178l9-1.mq.us-east-2.amazonaws.com:61617,
                                        ssl://
b-9876l5k4-32ji-109h-8gfe-7d65c4b132a1-2.mq.us-west-2.amazonaws.com:61617)"/>
                                    </networkConnectors>
```

> **Note**
>
> Kami merekomendasikan penggunaan prefiks `masterslave:` untuk jaringan broker. Prefiks identik dengan yang lebih sintaks `static:failover:()?randomize=false&maxReconnectAttempts=0` yang lebih eksplisit.

> **Note**
>
> Konfigurasi XHTML ini tidak mengizinkan spasi.

kahaDB

`kahaDB` adalah anak dari elemen pengumpulan anak `persistenceAdapter`.

Atribut

concurrentStoreAndDispatchQueues

Menentukan apakah akan menggunakan penyimpanan bersamaan dan pengiriman untuk antrean. Untuk informasi selengkapnya, lihat Menonaktifkan Penyimpanan dan Pengiriman Bersamaan untuk Antrean dengan Konsumen Lambat.

Default: `true`

cleanupOnStop

> Didukung di
>
> Apache ActiveMQ 15.16.x dan yang lebih baru

Jika dinonaktifkan, pengumpulan dan pembersihan sampah tidak terjadi ketika broker dihentikan, yang mempercepat proses shutdown. Peningkatan kecepatan berguna dalam kasus dengan basis data besar atau basis data penjadwal.

Default: `true`

journalDiskSyncInterval

Interval (mdtk) untuk kapan harus melakukan sinkronisasi disk jika `journalDiskSyncStrategy=periodic`. Untuk informasi selengkapnya, lihat dokumentasi Apache ActiveMQ kahaDB .

Default: `1000`

journalDiskSyncStrategi

> Didukung di
>
> Apache ActiveMQ 15.14.x dan yang lebih baru

Mengonfigurasi kebijakan sinkronisasi disk. Untuk informasi selengkapnya, lihat dokumentasi Apache ActiveMQ kahaDB .

Default: `always`

> **Note**
>
> Dokumentasi ActiveMQ menyatakan bahwa kehilangan data dibatasi dalam durasi `journalDiskSyncInterval`, yang memiliki default 1dtk. Kehilangan data bisa lebih lama dibandingkan interval, namun sulit untuk menemukan durasi yang tepat. Berhati-hatilah.

preallocationStrategy

Mengonfigurasi cara broker akan melakukan pra-alokasi file jurnal ketika file jurnal baru diperlukan. Untuk informasi selengkapnya, lihat dokumentasi Apache ActiveMQ kahaDB .

Default: `sparse_file`

Contoh Konfigurasi

Example

```
<broker xmlns="http://activemq.apache.org/schema/core">
                                <persistenceAdapter>
                                <kahaDB preallocationStrategy="zeros"
 concurrentStoreAndDispatchQueues="false" journalDiskSyncInterval="10000"
 journalDiskSyncStrategy="periodic"/>
                             </persistenceAdapter>
                             </broker>
```

systemUsage

`systemUsage` adalah anak dari elemen pengumpulan anak `systemUsage`. Ini mengontrol jumlah maksimum ruang yang akan digunakan broker sebelum memperlambat produsen. Untuk informasi selengkapnya, lihat Kontrol Alur Produsen dalam dokumentasi Apache ActiveMQ.

Elemen Anak

memoryUsage

`memoryUsage` adalah anak dari elemen anak `systemUsage`. Ini mengelola penggunaan memori. Gunakan `memoryUsage` untuk melacak durasi penggunaan sesuatu sehingga Anda dapat mengontrol penggunaan set kerja secara produktif. Untuk informasi selengkapnya, lihat skema dalam dokumentasi Apache ActiveMQ.

Elemen Anak

`memoryUsage` adalah anak dari elemen anak `memoryUsage`.

Atribut

percentOfJvmTumpukan

Integer antara 0 (inklusif) dan 70 (inklusif).

Default: `70`

Atribut

sendFailIfNoSpace

Menetapkan apakah metode `send()` harus gagal jika tidak ada ruang kosong. Nilai default adalah false, yang memblokir metode `send()` hingga ruang menjadi tersedia. Untuk informasi selengkapnya, lihat skema dalam dokumentasi Apache ActiveMQ.

Default: `false`

sendFailIfNoSpaceAfterTimeout

Default: `null`

Contoh Konfigurasi

Example

```
<broker xmlns="http://activemq.apache.org/schema/core">
                                <systemUsage>
                                <systemUsage sendFailIfNoSpace="true"
 sendFailIfNoSpaceAfterTimeout="2000">
                                    <memoryUsage>
                                        <memoryUsage  percentOfJvmHeap="60" />
```

```
                    </memoryUsage>>
                </systemUsage>
                </systemUsage>
            </broker>
            </persistenceAdapter>
```

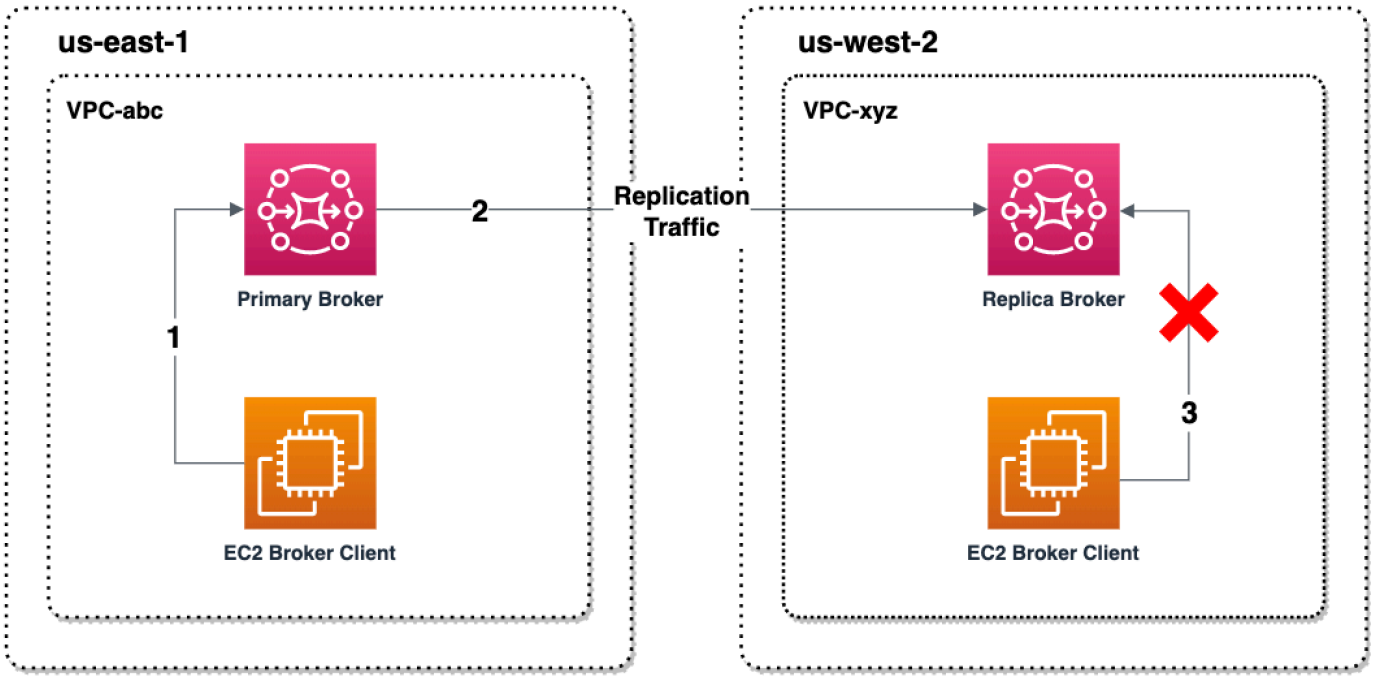
Replikasi data lintas wilayah untuk Amazon MQ untuk ActiveMQ

Amazon MQ untuk ActiveMQ menawarkan fitur replikasi data Lintas Wilayah (CRDR) yang memungkinkan replikasi pesan asinkron dari broker utama di Wilayah utama ke broker replika di Wilayah replika. AWS Dengan mengeluarkan permintaan failover ke Amazon MQ API, broker replika saat ini dipromosikan ke peran broker utama, dan broker utama saat ini diturunkan ke peran replika.

Broker primer dan replika untuk replikasi data lintas wilayah

Anda dapat membuat broker primer dan replika untuk replikasi data asinkron dari broker utama di Wilayah utama ke broker replika di AWS Wilayah replika. Wilayah utama terdiri dari sepasang broker aktif/siaga yang berlebihan yang disebut sebagai broker utama. Wilayah sekunder terdiri dari sepasang broker aktif/siaga yang berlebihan yang disebut sebagai broker replika.

Diagram berikut menggambarkan broker replika di Wilayah sekunder yang menerima data replikasi asinkron dari broker utama di Wilayah primer.



Pialang primer dan replika bertindak sebagai solusi pemulihan data lintas wilayah. Jika broker utama di Wilayah primer gagal, Anda dapat mempromosikan broker replika di Wilayah sekunder ke primer dengan memulai peralihan atau failover. Mantan broker utama kemudian menjadi broker replika, dan mantan broker replika dipromosikan menjadi broker utama. Untuk petunjuk tentang membuat broker utama dan replika, lihatMembuat broker replikasi data lintas wilayah Amazon MQ.

> **Note**
>
> Hanya tersedia untuk broker aktif/siaga.
> Tidak tersedia untuk antrian cermin.

Membuat broker replikasi data lintas wilayah Amazon MQ

Dengan replikasi data Lintas Wilayah (CRDR), Anda dapat beralih antara Amazon MQ untuk broker pesan ActiveMQ di dua Wilayah AWS sesuai kebutuhan. Anda dapat menunjuk broker yang sudah ada sebagai broker utama dan membuat replika untuk broker ini, atau membuat broker primer dan replika baru bersama-sama. Anda kemudian dapat mempromosikan broker replika ke peran broker utama menggunakan operasi Amazon `Promote` MQ API. Untuk informasi lebih lanjut tentang broker primer dan replika, lihatBroker primer dan replika untuk replikasi data lintas wilayah.

Petunjuk berikut menjelaskan bagaimana Anda dapat membuat dan mengonfigurasi broker replika menggunakan Amazon MQ Management Console.

Topik

- Prasyarat
- Langkah 1 (Opsional): Buat broker utama baru
- Langkah 2: Buat replika broker yang ada

Prasyarat

Untuk menggunakan fitur replikasi data lintas wilayah, Anda harus meninjau dan mematuhi prasyarat berikut:

- Versi: Fitur replikasi data lintas wilayah hanya tersedia untuk Amazon MQ untuk broker ActiveMQ pada versi 5.17.6 ke atas.

- Wilayah: Replikasi data Lintas Wilayah didukung di wilayah berikut: AS Timur (Ohio), AS Timur (Virginia N.), AS Barat (Oregon), dan AS Barat (California N.).
- Jenis instans: Replikasi data Lintas Wilayah hanya tersedia untuk ukuran `mq.m5.large` instans broker ke atas.
- Jenis penyebaran: Replikasi data Lintas Wilayah hanya tersedia untuk broker aktif/siaga dengan penyebaran zona multi-ketersediaan.
- Status broker: Anda hanya dapat membuat broker replika untuk broker utama dengan status `Running` broker.

Langkah 1 (Opsional): Buat broker utama baru

Buat broker utama baru

1. Masuk ke konsol Amazon MQ.
2. Pada halaman Broker konsol Amazon MQ, pilih Buat broker.
3. Di halaman Pilih mesin broker, pilih Apache ActiveMQ.
4. Di halaman Pilih deployment dan penyimpanan, pada bagian Mode deployment dan jenis penyimpanan, lakukan hal berikut:
    - Untuk mode Deployment, pilih Active/Standby broker. Broker aktif/siaga terdiri dari dua broker di dua Availability Zone berbeda yang dikonfigurasi dalam pasangan redundan. Broker ini berkomunikasi secara serempak dengan aplikasi Anda dan dengan Amazon EFS. Untuk informasi selengkapnya, lihat Opsi penyebaran untuk Amazon MQ untuk broker ActiveMQ.
5. Pilih Berikutnya.
6. Di halaman Konfigurasi pengaturan, pada bagian Detail, lakukan hal berikut:
    a. Masukkan nama Broker.

> **Important**
>
> Jangan menambahkan informasi identitas pribadi (PII) atau informasi rahasia atau sensitif lainnya dalam nama broker. Nama broker dapat diakses oleh AWS layanan lain, termasuk CloudWatch Log. Nama broker tidak dimaksudkan untuk digunakan untuk data pribadi atau sensitif.

b. Pilih Tipe instans broker (misalnya, mq.m5.large). Untuk informasi selengkapnya, lihat Broker instance types.

7. Di bagian Akses Konsol Web ActiveMQ, sediakan Nama pengguna dan Kata sandi. Pembatasan berikut berlaku untuk nama pengguna dan kata sandi broker:

   - Nama pengguna Anda hanya dapat berisi karakter alfanumerik, tanda hubung, titik, garis bawah, dan tildes (-. _ ~).
   - Kata sandi Anda setidaknya harus terdiri dari 12 karakter, berisi setidaknya 4 karakter unik, dan tidak boleh berisi koma, titik dua, atau tanda yang sama (,:=).

> **Important**
>
> Jangan menambahkan informasi identitas pribadi (PII) atau informasi rahasia atau sensitif lainnya dalam nama pengguna broker. Nama pengguna broker dapat diakses oleh AWS layanan lain, termasuk CloudWatch Log. Nama pengguna broker tidak dimaksudkan untuk digunakan untuk data pribadi atau sensitif.

Bilah lampu kilat hijau di bagian atas halaman mengonfirmasi bahwa Amazon MQ membuat broker replika di Wilayah pemulihan. Anda juga dapat melihat peran CRDR dan status RPO untuk broker Anda. Untuk mematikan kolom Peran CRDR dan Status RPO, pilih ikon roda gigi di sudut kanan atas tabel Broker. Kemudian, pada halaman Preferensi, matikan Peran CRDR atau Status RPO.

Langkah 2: Buat replika broker yang ada

1. Pada halaman Broker konsol Amazon MQ, pilih Buat broker replika.
2. Pada halaman Pilih broker utama, pilih broker yang ada untuk digunakan sebagai broker utama CRDR. Lalu, pilih Selanjutnya.
3. Pada halaman Configure replica broker, gunakan menu drop-down untuk memilih Region replika.
4. Di bagian pengguna konsol ActiveMQ untuk broker replika, berikan Nama Pengguna dan Kata Sandi untuk pengguna konsol broker replika. Pembatasan berikut berlaku untuk nama pengguna dan kata sandi broker:

   - Nama pengguna Anda hanya dapat berisi karakter alfanumerik, tanda hubung, titik, garis bawah, dan tildes (-. _ ~).

- Kata sandi Anda setidaknya harus terdiri dari 12 karakter, berisi setidaknya 4 karakter unik, dan tidak boleh berisi koma, titik dua, atau tanda yang sama (,:=).

> **Important**
>
> Jangan menambahkan informasi identitas pribadi (PII) atau informasi rahasia atau sensitif lainnya dalam nama pengguna broker. Nama pengguna broker dapat diakses oleh AWS layanan lain, termasuk CloudWatch Log. Nama pengguna broker tidak dimaksudkan untuk digunakan untuk data pribadi atau sensitif.

5. Pada pengguna replikasi Data untuk menjembatani akses antar broker, berikan Username dan Password bagi pengguna yang akan mengakses broker primer dan replika. Pembatasan berikut berlaku untuk nama pengguna dan kata sandi broker:

   - Nama pengguna Anda hanya dapat berisi karakter alfanumerik, tanda hubung, titik, garis bawah, dan tildes (-. _ ~).
   - Kata sandi Anda setidaknya harus terdiri dari 12 karakter, berisi setidaknya 4 karakter unik, dan tidak boleh berisi koma, titik dua, atau tanda yang sama (,:=).

   > **Important**
   >
   > Jangan menambahkan informasi identitas pribadi (PII) atau informasi rahasia atau sensitif lainnya dalam nama pengguna broker. Nama pengguna broker dapat diakses oleh AWS layanan lain, termasuk CloudWatch Log. Nama pengguna broker tidak dimaksudkan untuk digunakan untuk data pribadi atau sensitif.

   Konfigurasikan pengaturan tambahan apa pun. Lalu, pilih Selanjutnya.

6. Pada halaman Tinjau dan buat, tinjau detail broker replika. Kemudian, pilih Buat broker replika.
7. Selanjutnya, reboot broker utama. Ini juga akan me-reboot broker replika. Untuk petunjuk tentang me-reboot broker Anda, lihat. Rebooting a Broker

Untuk informasi selengkapnya tentang mengonfigurasi pengaturan tambahan untuk broker ActiveMQ Anda, lihat Memulai: Membuat dan menghubungkan ke broker ActiveMQ

Menghapus broker replikasi data lintas wilayah Amazon MQ

Untuk menghapus broker replikasi data lintas wilayah (CRDR) primer atau replika, Anda harus terlebih dahulu memutuskan pasangan kemudian me-reboot broker. Petunjuk berikut menunjukkan bagaimana Anda dapat memutuskan pasangan dan me-reboot broker menggunakan Konsol AWS Manajemen.

1. Pada halaman Broker, pilih broker CRDR yang ingin Anda hapus pasangannya, lalu pilih Edit.
2. Pada halaman Edit broker di bagian Replikasi data, pilih Unpair broker.
3. Masukkan “konfirmasi” di jendela pop-up untuk mengonfirmasi pilihan Anda. Kemudian pilih Unpair broker.
4. Selanjutnya, reboot broker utama yang tidak berpasangan. Ini juga akan me-reboot broker replika. Untuk petunjuk tentang me-reboot broker Anda, lihat. Rebooting a Broker Setelah broker utama di-boot ulang, kedua broker tidak berpasangan dan dapat dihapus secara individual. Untuk menghapus broker Anda, lihatDeleting a broker.

Memulai switchover atau failover untuk mempromosikan broker replika Amazon MQ ke peran broker utama

Anda dapat memulai switchover atau failover ketika Anda ingin mempromosikan broker replika ke peran broker utama. Ketika Anda mempromosikan broker replika, broker utama diturunkan ke peran broker replika.

Peralihan memprioritaskan konsistensi daripada ketersediaan. Pialang dijamin memiliki status yang sama ketika operasi failover ini selesai. Dengan peralihan, mungkin ada periode di mana tidak ada broker yang tersedia untuk koneksi klien sementara konsistensi antar-broker ditetapkan. Kedua broker akan memiliki status yang sama pada saat replika dipromosikan. Keberhasilan peralihan tergantung pada kesehatan kedua wilayah dan jaringan antar wilayah untuk berhasil.

Failover memprioritaskan ketersediaan daripada konsistensi. Pialang tidak dijamin memiliki status yang identik ketika operasi ini selesai. Dengan failover, broker replika dijamin akan segera tersedia untuk melayani lalu lintas klien, tanpa menunggu data replikasi apa pun disinkronkan, atau yang utama menerima sinyal shutdown. Failover tidak bergantung pada kesehatan wilayah primer asli maupun jaringan antar wilayah untuk berhasil.

Diagram berikut menggambarkan peralihan di mana tidak ada broker yang menerima koneksi klien saat antrian replikasi sedang dikeringkan dan status broker disinkronkan. Dalam proses ini, klien

di VPC broker utama tidak dapat menghasilkan perubahan status lebih lanjut saat operasi sedang berlangsung, dan broker utama diturunkan ke replika. Ketika antrian replikasi terkuras dan kedua broker mencapai keadaan yang sama, klien di VPC broker replika tidak dapat terhubung ke broker replika sampai operasi failover selesai, dan broker replika dipromosikan ke primer.

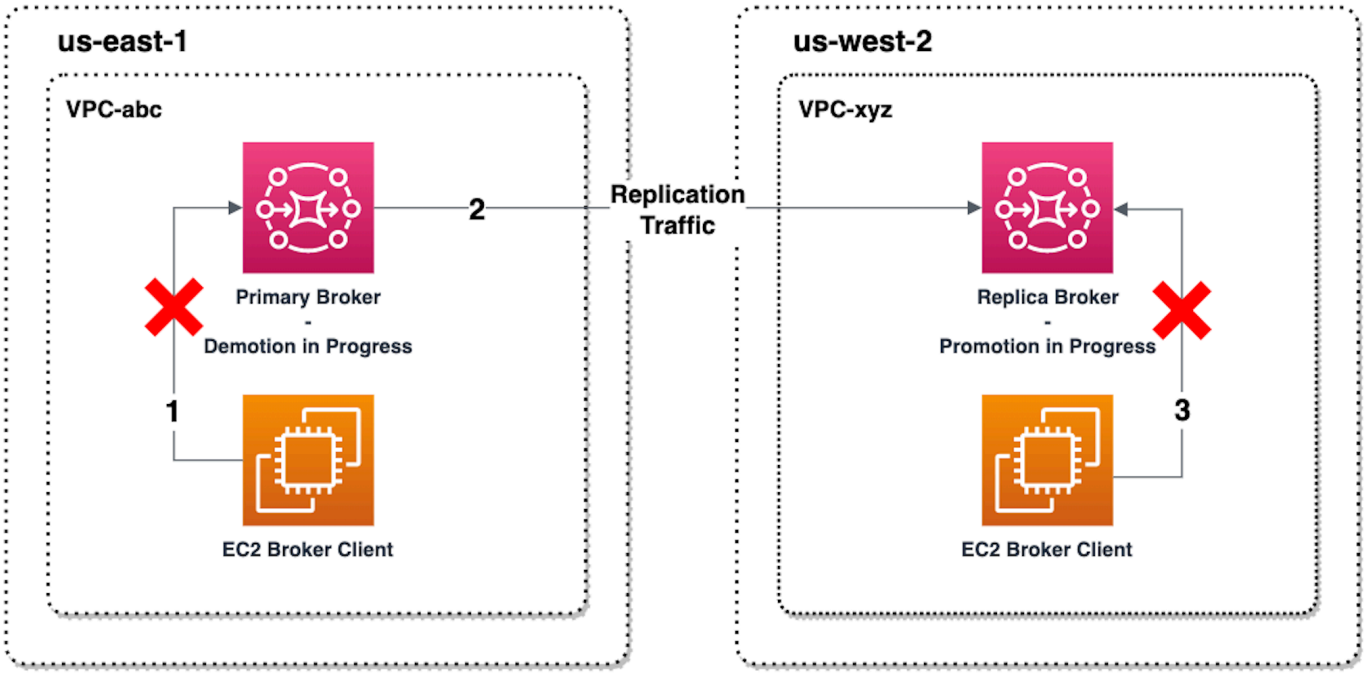
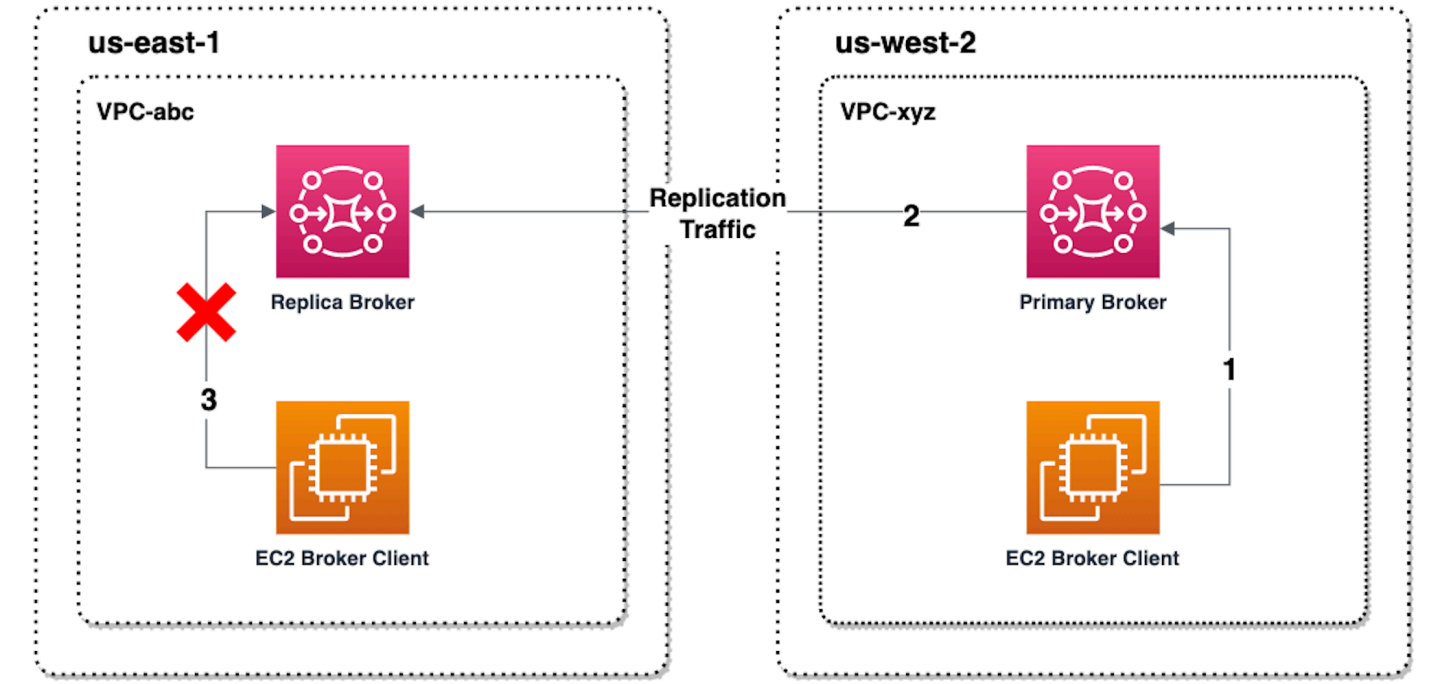


Diagram berikut menggambarkan status broker setelah proses peralihan selesai. Broker replika asli sekarang telah dipromosikan ke peran broker utama dan menerima koneksi klien. Klien dapat memproduksi dan mengkonsumsi data dari broker.



Promosikan broker replika menggunakan konsol

Untuk mempromosikan broker replika menggunakan switchover atau failover, ikuti langkah-langkah ini di konsol Amazon MQ.

> **Note**
>
> Anda tidak dapat memulai switchover atau failover pada broker utama.

1. Beralih ke wilayah untuk broker replika Anda. Dari tabel Broker Anda, pilih broker replika yang ada yang akan Anda promosikan ke primer.
2. Pada halaman detail Broker, lakukan hal berikut:
    1. Pilih Promosikan replika.
    2. Di jendela pop up, pilih Switchover atau Failover.
    3. Ketik “konfirmasi” di kotak teks untuk mengonfirmasi pilihan Anda.
    4. Pilih Konfirmasi.

Setelah memulai failover, status broker berubah menjadi Failover yang sedang berlangsung. Bilah kemajuan biru di bagian atas halaman Broker menjadi hijau saat failover selesai.

> **Note**
>
> Konfigurasi hanya direplikasi pada saat broker replika dibuat. Pembaruan apa pun setelahnya tidak direplikasi.

Metrik replikasi data lintas wilayah di Amazon CloudWatch

Fitur replikasi data lintas wilayah Amazon MQ untuk ActiveMQ menawarkan metrik untuk menjaga keandalan, ketersediaan, dan kinerja broker utama dan replika Anda. Selama proses replikasi, broker replika di Wilayah sekunder menerima data yang direplikasi secara asinkron dari broker utama di Wilayah primer. Jika broker utama di Wilayah primer gagal, Anda dapat mempromosikan broker replika di Wilayah sekunder ke primer dengan memulai peralihan atau failover. Untuk petunjuk tentang melihat metrik di Amazon CloudWatch, lihatMengakses CloudWatch metrik untuk Amazon MQ.

Cap waktu CRDR

Stempel waktu berikut menjelaskan bagaimana metrik yang ditemukan di Amazon CloudWatch dihitung. Ada lima stempel waktu dalam proses replikasi data:

- Waktu pengamatan saat ini (TCO): Waktu saat ini dalam waktu.
- Time of Creation (TC): Seketika suatu peristiwa dibuat pada antrian replikasi oleh broker utama. Tersedia di broker primer dan replika.
- Waktu pengiriman (TD): Seketika suatu acara berhasil dikirim ke broker replika. Hanya tersedia di broker replika.
- Waktu pemrosesan (TP): Seketika suatu peristiwa berhasil diproses oleh broker replika. Hanya tersedia di broker replika.
- Waktu pengakuan (TA): Seketika suatu peristiwa berhasil diakui oleh broker utama. Hanya tersedia di broker utama.

Perkirakan kinerja switchover/failover dengan metrik CRDR CloudWatch

Amazon MQ memungkinkan metrik untuk broker Anda secara default. Anda dapat melihat metrik broker Anda dengan mengakses CloudWatch konsol Amazon, atau dengan menggunakan API. CloudWatch Metrik berikut berguna untuk memahami replikasi dan kinerja switchover/failover broker CRDR Anda:

Metrik Amazon MQ CloudWatch	Alasan penggunaan CRDR
`TotalReplicationLag`	Perkiraan waktu antara TA dan TC dari peristiwa terakhir yang tidak diakui di broker utama.
`ReplicationLag`	Perkiraan waktu antara TP dan TC dari peristiwa terakhir yang tidak diakui di broker replika.
`PrimaryWaitTime`	Perkiraan waktu antara TCO dan TC dari acara yang diproses terakhir pada broker utama.
`ReplicaWaitTime`	Perkiraan waktu antara TCO dan TP dari acara yang diproses terakhir pada broker replika.
`QueueSize`	Jumlah total peristiwa yang tidak diakui dalam antrian replikasi pada broker utama.

`TotalReplicationLag`dan `ReplicationLag` menggambarkan replikasi yang tertunda antara broker primer dan replika. Kedua metrik juga dapat digunakan untuk memperkirakan waktu hingga operasi switchover atau failover yang sedang berlangsung selesai.

`PrimaryWaitTime`dan `ReplicaWaitTime` dapat digunakan untuk mengidentifikasi masalah yang sedang berlangsung dengan proses replikasi. Jika nilai metrik terus bertambah, ini dapat menunjukkan proses replikasi terdegradasi atau dijeda. Replikasi lambat dapat terjadi karena masalah seperti partisi jaringan, mulai broker, dan pemulihan yang lama.

Tutorial ActiveMQ

Tutorial berikut menunjukkan cara membuat dan terhubung ke broker ActiveMQ Anda. Untuk menggunakan kode contoh ActiveMQ Java, Anda harus menginstal Kit Pengembangan Java Standard Edition dan membuat beberapa perubahan pada kode

Topik

- Membuat dan mengonfigurasi jaringan broker Amazon MQ
- Menghubungkan aplikasi Java ke broker Amazon MQ Anda
- Mengintegrasikan broker ActiveMQ dengan LDAP
- Langkah 3: (Opsional) Connect ke AWS Lambda fungsi
- Membuat pengguna broker ActiveMQ
- Edit pengguna broker ActiveMQ
- Hapus pengguna broker ActiveMQ
- Contoh kerja menggunakan Java Message Service (JMS) dengan ActiveMQ

Membuat dan mengonfigurasi jaringan broker Amazon MQ

Jaringan broker terdiri dari beberapa broker instans tunggal aktif atau broker aktif/siaga. Dalam tutorial ini, Anda mempelajari cara membuat jaringan broker dua broker dengan topologi source and sink.

Untuk gambaran umum konseptual dan detail informasi konfigurasi, lihat hal berikut:

- Jaringan broker Amazon MQ
- Mengonfigurasi Jaringan Broker dengan Benar
- `networkConnector`
- `networkConnectionStartAsinkron`
- Jaringan Broker dalam dokumentasi ActiveMQ

Anda dapat menggunakan konsol Amazon MQ untuk membuat jaringan broker Amazon MQ. Karena Anda dapat memulai pembuatan dua broker secara paralel, proses ini berdurasi sekitar 15 menit.

Topik

- Prasyarat
- Langkah 1: Mengizinkan Lalu Lintas antara Broker
- Langkah 2: Mengonfigurasi Konektor Jaringan untuk Broker Anda
- Langkah selanjutnya

Prasyarat

Untuk membuat jaringan broker, Anda harus memiliki hal berikut:

- Dua atau lebih broker aktif secara bersamaan (bernama `MyBroker1` dan `MyBroker2` dalam tutorial ini). Untuk informasi selengkapnya tentang cara membuat broker, lihat Memulai: Membuat dan menghubungkan ke broker ActiveMQ.
- Kedua broker harus berada di VPC yang sama atau di peered. VPCs Untuk informasi selengkapnya VPCs, lihat Apa itu Amazon VPC? di Panduan Pengguna Amazon VPC dan Apa itu VPC Peering? di Panduan Peering VPC Amazon.

  > **Important**
  >
  > Jika tidak memiliki VPC, subnet, atau grup keamanan default, Anda harus membuatnya terlebih dahulu. Untuk informasi selengkapnya, lihat hal berikut dalam Panduan Pengguna Amazon VPC:
  >
  > - Membuat VPC Default
  > - Membuat Subnet Default
  > - Membuat Grup Keamanan

- Dua pengguna dengan kredensi masuk yang identik untuk kedua broker. Untuk informasi lebih lanjut tentang membuat pengguna, lihat Membuat pengguna broker ActiveMQ.

  > **Note**
  >
  > Ketika mengintegrasikan autentikasi LDAP dengan jaringan broker, pastikan bahwa pengguna ada sebagai broker ActiveMQ, serta pengguna LDAP.

Contoh berikut menggunakan dua broker instans tunggal. Namun, Anda dapat membuat jaringan broker menggunakan broker aktif/siaga atau kombinasi mode deployment broker.

Langkah 1: Mengizinkan Lalu Lintas antara Broker

Setelah membuat broker, Anda harus mengizinkan lalu lintas di antara broker.

1. Di konsol Amazon MQ, di halaman MyBroker2, di bagian Detail, di bawah Keamanan dan jaringan, pilih nama grup keamanan Anda atau.

   Halaman Grup Keamanan Dasbor EC2 akan ditampilkan.
2. Dari daftar grup keamanan, pilih grup keamanan Anda.
3. Di bagian bawah halaman, pilih tab Masuk, lalu pilih Edit.
4. Dalam kotak dialog Edit aturan masuk, tambahkan aturan untuk titik OpenWire akhir.

   a. Pilih Tambahkan aturan.

   b. Untuk Jenis, pilih TCP Kustom.

   c. Untuk Port Range, ketik OpenWire port (`61617`).

   d. Lakukan salah satu tindakan berikut:

   - Jika Anda ingin membatasi akses ke alamat IP tertentu, untuk Sumber, biarkan Khusus dipilih, lalu masukkan alamat IP `MyBroker1`, diikuti oleh `/32`. (Ini mengubah alamat IP menjadi catatan CIDR yang valid). Untuk informasi selengkapnya, lihat Antarmuka Jaringan Elastis.

     > **Tip**
     >
     > Untuk mengambil alamat IP `MyBroker1`, di Konsol Amazon MQ, pilih nama broker dan arahkan ke bagian Detail.

   - Jika semua broker adalah privat dan berada di VPC yang sama, untukSumber, biarkan Khusus dipilih, lalu ketik ID grup keamanan yang Anda edit.

     > **Note**
     >
     > Untuk broker publik, Anda harus membatasi akses menggunakan alamat IP.

e. Pilih Simpan.

Broker Anda kini dapat menerima koneksi masuk.

Langkah 2: Mengonfigurasi Konektor Jaringan untuk Broker Anda

Setelah Anda mengizinkan lalu lintas di antara broker, Anda harus mengonfigurasi konektor jaringan untuk salah satu broker.

1. Mengedit revisi konfigurasi untuk broker `MyBroker1`.

   a. Pada halaman MyBroker1, pilih Edit.

   b. Pada halaman Edit MyBroker 1, di bagian Konfigurasi, pilih Lihat.

   Jenis dan versi mesin broker yang digunakan konfigurasi (misalnya, Apache ActiveMQ 5.15.0) akan ditampilkan.

   c. Di tab Detail konfigurasi, nomor revisi konfigurasi, deskripsi, dan konfigurasi broker dalam format XML akan ditampilkan.

   d. Pilih Edit konfigurasi.

   e. Di bagian bawah file konfigurasi, hapus komentar bagian `<networkConnectors>` dan sertakan informasi berikut:

   - `name` untuk konektor jaringan.
   - `username` Konsol Web ActiveMQ yang umum untuk kedua broker.
   - Mengaktifkan koneksi `duplex`.
   - Lakukan salah satu tindakan berikut:
     - Jika Anda menghubungkan broker ke broker satu contoh, gunakan `static:` awalan dan titik OpenWire akhir `uri` untuk. `MyBroker2` Contoh:

```
<networkConnectors>
  <networkConnector name="connector_1_to_2" userName="myCommonUser"
 duplex="true"
    uri="static:(ssl://b-1234a5b6-78cd-901e-2fgh-3i45j6k178l9-1.mq.us-
east-2.amazonaws.com:61617)"/>
</networkConnectors>
```

- Jika Anda menghubungkan broker ke broker aktif/siaga, gunakan `static+failover` transportasi dan OpenWire titik akhir `uri` untuk kedua broker dengan parameter kueri berikut. `?randomize=false&maxReconnectAttempts=0` Contoh:

```
<networkConnectors>
  <networkConnector name="connector_1_to_2" userName="myCommonUser"
 duplex="true"
    uri="static:(failover:(ssl://
b-1234a5b6-78cd-901e-2fgh-3i45j6k178l9-1.mq.us-east-2.amazonaws.com:61617,
    ssl://b-9876l5k4-32ji-109h-8gfe-7d65c4b132a1-2.mq.us-
west-2.amazonaws.com:61617)?randomize=false&amp;maxReconnectAttempts=0)"/>
</networkConnectors>
```

> **Note**
>
> Jangan sertakan kredenal masuk untuk pengguna ActiveMQ.

f. Pilih Simpan.

g. Di kotak dialog Simpan revisi, ketik `Add network of brokers connector for MyBroker2`.

h. Pilih Simpan untuk menyimpan revisi konfigurasi baru.

2. Edit `MyBroker1` untuk mengatur revisi konfigurasi terbaru agar segera diterapkan.

a. Pada halaman MyBroker1, pilih Edit.

b. Pada halaman Edit MyBroker 1, di bagian Konfigurasi, pilih Jadwal Modifikasi.

c. Pada bagian Jadwalkan perubahan broker, pilih terapkan perubahan Segera.

d. Pilih Terapkan .

`MyBroker1` akan di-boot ulang dan revisi konfigurasi Anda diterapkan.

Jaringan broker dibuat.

Langkah selanjutnya

Setelah mengonfigurasi jaringan broker, Anda dapat mengujinya dengan memproduksi dan mengonsumsi pesan.

> **Important**
>
> Pastikan Anda mengaktifkan koneksi masuk dari mesin lokal Anda untuk broker `MyBroker1` di port 8162 (untuk ActiveMQ Web Console) dan port 61617 (untuk titik akhir). OpenWire Anda mungkin juga perlu menyesuaikan pengaturan grup keamanan agar produsen dan konsumen dapat terhubung ke jaringan broker.

1. Pada konsol Amazon MQ, arahkan ke bagian Koneksi dan perhatikan titik akhir Konsol Web ActiveMQ untuk broker `MyBroker1`.
2. Arahkan ke Konsol Web ActiveMQ untuk broker `MyBroker1`.
3. Untuk memverifikasi bahwa jembatan jaringan sudah terhubung, pilih Jaringan.

   Di Jembatan Jaringan, nama dan alamat `MyBroker2` tercantum dalam kolom Broker Jarak Jauh dan Alamat Jarak Jauh.
4. Dari setiap mesin yang memiliki akses ke broker `MyBroker2`, buat konsumen. Contoh:

   ```
   activemq consumer --brokerUrl "ssl://
   b-1234a5b6-78cd-901e-2fgh-3i45j6k178l9-1.mq.us-east-2.amazonaws.com:61617" \
    --user commonUser \
    --password myPassword456 \
    --destination queue://MyQueue
   ```

   Konsumen terhubung ke OpenWire titik akhir `MyBroker2` dan mulai mengkonsumsi pesan dari antrian`MyQueue`.
5. Dari setiap mesin yang memiliki akses ke broker `MyBroker1`, buat produsen dan kirimkan beberapa pesan. Contoh:

   ```
   activemq producer --brokerUrl "ssl://
   b-9876l5k4-32ji-109h-8gfe-7d65c4b132a1-1.mq.us-east-2.amazonaws.com:61617" \
    --user commonUser \
    --password myPassword456 \
    --destination queue://MyQueue \
    --persistent true \
    --messageSize 1000 \
    --messageCount 10000
   ```

Produser terhubung ke OpenWire titik akhir `MyBroker1` dan mulai menghasilkan pesan persisten ke antrian`MyQueue`.

Menghubungkan aplikasi Java ke broker Amazon MQ Anda

Setelah membuat broker ActiveMQ Amazon MQ, Anda dapat menghubungkan aplikasi ke broker. Contoh berikut menunjukkan cara menggunakan Layanan Pesan Java (JMS) untuk membuat koneksi ke broker, membuat antrean, dan mengirim pesan. Untuk contoh Java yang lengkap dapat berfungsi, lihat Working Java Example.

Anda dapat terhubung ke broker ActiveMQ menggunakan berbagai klien ActiveMQ. Kami merekomendasikan penggunaan Klien ActiveMQ.

Topik

- Prasyarat
- Untuk Membuat Produsen Pesan dan Mengirimkan Pesan
- Untuk Membuat Konsumen Pesan dan Menerima Pesan

Prasyarat

Mengaktifkan Atribut VPC

Untuk memastikan bahwa broker dapat diakses dalam VPC, Anda harus mengaktifkan atribut VPC `enableDnsHostnames` dan `enableDnsSupport`. Untuk informasi selengkapnya, lihat Dukungan DNS di VPC Anda dalam Panduan Pengguna Amazon VPC.

Mengaktifkan Koneksi Masuk

Selanjutnya, aktifkan koneksi masuk untuk aplikasi Anda.

1. Masuk ke konsol Amazon MQ.
2. Dari daftar broker, pilih nama broker Anda (misalnya, MyBroker).
3. Pada ***`MyBroker`***halaman, di bagian Koneksi, catat alamat dan port URL konsol web broker dan protokol tingkat kabel.
4. Di bagian Detail, di bawah Keamanan dan jaringan, pilih nama grup keamanan Anda atau

Halaman Grup Keamanan Dasbor EC2 akan ditampilkan.

5. Dari daftar grup keamanan, pilih grup keamanan Anda.
6. Di bagian bawah halaman, pilih tab Masuk, lalu pilih Edit.
7. Di kotak dialog Edit aturan masuk, tambahkan aturan untuk setiap URL atau titik akhir yang Anda inginkan untuk dapat diakses secara publik (contoh berikut menampilkan cara melakukannya untuk konsol web broker).

   a. Pilih Tambahkan aturan.

   b. Untuk Jenis, pilih TCP Kustom.

   c. Untuk Rentang Port, ketik port konsol web (`8162`).

   d. Untuk Sumber, biarkan Kustom dipilih lalu ketik alamat IP sistem yang Anda inginkan untuk dapat mengakses konsol web (misalnya, `192.0.2.1`).

   e. Pilih Simpan.

      Broker Anda kini dapat menerima koneksi masuk.

Menambahkan Dependensi Java

Tambahkan paket `activemq-client.jar` dan `activemq-pool.jar` ke jalur kelas Java Anda. Contoh berikut menampilkan dependensi ini dalam file `pom.xml` proyek Maven.

```
<dependencies>
    <dependency>
        <groupId>org.apache.activemq</groupId>
        <artifactId>activemq-client</artifactId>
        <version>5.15.16</version>
    </dependency>
    <dependency>
        <groupId>org.apache.activemq</groupId>
        <artifactId>activemq-pool</artifactId>
        <version>5.15.16</version>
    </dependency>
</dependencies>
```

Untuk informasi selengkapnya tentang `activemq-client.jar`, lihat Konfigurasi Awaldalam dokumentasi Apache ActiveMQ.

> **Important**
>
> Pada kode contoh berikut, produsen dan konsumen berjalan dalam satu utas. Untuk sistem produksi (atau untuk menguji failover instans broker), pastikan bahwa produsen dan konsumen berjalan di host atau utas terpisah.

Untuk Membuat Produsen Pesan dan Mengirimkan Pesan

Gunakan instruksi berikut untuk membuat produser pesan dan menerima pesan.

1. Membuat pabrik koneksi yang dikumpulkan JMS untuk produsen pesan menggunakan titik akhir broker lalu memanggil metode `createConnection` untuk pabrik.

   > **Note**
   >
   > Untuk active/standby broker, Amazon MQ menyediakan dua ActiveMQ Web Console URLs, tetapi hanya satu URL yang aktif pada satu waktu. Demikian juga, Amazon MQ menyediakan dua titik akhir untuk setiap protokol tingkat wire, tetapi hanya satu titik akhir aktif di setiap pasangan pada satu waktu. Sufiks `-1` dan `-2` menunjukkan pasangan redundan. Untuk informasi selengkapnya, lihat Opsi penyebaran untuk Amazon MQ untuk broker ActiveMQ).
   > Untuk titik akhir protokol tingkat kabel, Anda harus mengizinkan aplikasi Anda terhubung ke salah satu titik akhir dengan menggunakan Failover Transport.

   ```
   // Create a connection factory.
   final ActiveMQConnectionFactory connectionFactory = new
    ActiveMQConnectionFactory(wireLevelEndpoint);

   // Pass the sign-in credentials.
   connectionFactory.setUserName(activeMqUsername);
   connectionFactory.setPassword(activeMqPassword);

   // Create a pooled connection factory.
   final PooledConnectionFactory pooledConnectionFactory = new
    PooledConnectionFactory();
   pooledConnectionFactory.setConnectionFactory(connectionFactory);
   pooledConnectionFactory.setMaxConnections(10);
   ```

```
// Establish a connection for the producer.
final Connection producerConnection = pooledConnectionFactory.createConnection();
producerConnection.start();

// Close all connections in the pool.
pooledConnectionFactory.clear();
```

> **Note**
>
> Produsen pesan harus selalu menggunakan kelas `PooledConnectionFactory`. Untuk informasi selengkapnya, lihat Selalu Gunakan Pooling Koneksi.

2. Membuat sesi, antrean bernama `MyQueue`, dan produser pesan.

```
// Create a session.
final Session producerSession = producerConnection.createSession(false,
 Session.AUTO_ACKNOWLEDGE);

// Create a queue named "MyQueue".
final Destination producerDestination = producerSession.createQueue("MyQueue");

// Create a producer from the session to the queue.
final MessageProducer producer =
 producerSession.createProducer(producerDestination);
producer.setDeliveryMode(DeliveryMode.NON_PERSISTENT);
```

3. Membuat string pesan `"Hello from Amazon MQ!"` lalu mengirimkan pesan.

```
// Create a message.
final String text = "Hello from Amazon MQ!";
TextMessage producerMessage = producerSession.createTextMessage(text);

// Send the message.
producer.send(producerMessage);
System.out.println("Message sent.");
```

4. Membersihkan produsen.

```
producer.close();
producerSession.close();
```

```
producerConnection.close();
```

Untuk Membuat Konsumen Pesan dan Menerima Pesan

Gunakan instruksi berikut untuk membuat produser pesan dan menerima pesan.

1. Membuat pabrik koneksi JMS untuk produsen pesan menggunakan titik akhir broker lalu memanggil metode `createConnection` untuk pabrik.

```
// Create a connection factory.
final ActiveMQConnectionFactory connectionFactory = new
 ActiveMQConnectionFactory(wireLevelEndpoint);

// Pass the sign-in credentials.
connectionFactory.setUserName(activeMqUsername);
connectionFactory.setPassword(activeMqPassword);

// Establish a connection for the consumer.
final Connection consumerConnection = connectionFactory.createConnection();
consumerConnection.start();
```

> **Note**
>
> Konsumen pesan jangan pernah gunakan kelas `PooledConnectionFactory`. Untuk informasi selengkapnya, lihat Selalu Gunakan Pooling Koneksi.

2. Membuat sesi, antrean bernama `MyQueue`, dan konsumen pesan.

```
// Create a session.
final Session consumerSession = consumerConnection.createSession(false,
 Session.AUTO_ACKNOWLEDGE);

// Create a queue named "MyQueue".
final Destination consumerDestination = consumerSession.createQueue("MyQueue");

// Create a message consumer from the session to the queue.
final MessageConsumer consumer =
 consumerSession.createConsumer(consumerDestination);
```

3. Mulai menunggu pesan dan menerima pesan saat tiba.

```
// Begin to wait for messages.
final Message consumerMessage = consumer.receive(1000);

// Receive the message when it arrives.
final TextMessage consumerTextMessage = (TextMessage) consumerMessage;
System.out.println("Message received: " + consumerTextMessage.getText());
```

> **Note**
>
> Tidak seperti layanan AWS pesan (seperti Amazon SQS), konsumen selalu terhubung ke broker.

4. Menutup konsumen, sesi, dan koneksi.

```
consumer.close();
consumerSession.close();
consumerConnection.close();
```

Mengintegrasikan broker ActiveMQ dengan LDAP

> **Important**
>
> Amazon MQ tidak mendukung sertifikat server yang dikeluarkan oleh CA pribadi.

Anda dapat mengakses broker ActiveMQ menggunakan protokol berikut dengan TLS yang diaktifkan:

- AMQP
- MQTT
- MQTT lebih WebSocket
- OpenWire
- MENGINJAK
- STOMP berakhir WebSocket

Amazon MQ menawarkan pilihan antara autentikasi ActiveMQ native serta autentikasi dan otorisasi LDAP untuk mengelola izin pengguna. Untuk informasi tentang pembatasan yang berkaitan dengan nama pengguna dan kata sandi ActiveMQ, lihat Pengguna.

Untuk mengotorisasi pengguna dan grup ActiveMQ agar dapat bekerja dengan antrean dan topik, Anda harus mengedit konfigurasi broker. Amazon MQ menggunakan Plugin Autentikasi Sederhana ActiveMQ untuk membatasi baca dan tulis ke tujuan. Untuk informasi selengkapnya dan contoh tambahan, lihat Selalu konfigurasikan peta otorisasi dan `authorizationEntry`.

> **Note**
>
> Saat ini, Amazon MQ tidak mendukung Autentikasi Sertifikat Klien.

Topik

- Mengintegrasikan LDAP dengan ActiveMQ
- Prasyarat
- Memulai dengan LDAP
- Cara kerja integrasi LDAP

Mengintegrasikan LDAP dengan ActiveMQ

Anda dapat mengautentikasi pengguna Amazon MQ melalui kredensial yang disimpan dalam server Lightweight Directory Access Protocol (LDAP). Anda juga dapat menambahkan, menghapus, dan memodifikasi pengguna Amazon MQ serta menetapkan izin untuk topik juga antrean dari sana. Operasi manajemen seperti membuat, memperbarui, dan menghapus broker masih memerlukan kredensial IAM dan tidak terintegrasi dengan LDAP.

Pelanggan yang ingin menyederhanakan dan memusatkan autentikasi dan otorisasi broker Amazon MQ mereka menggunakan server LDAP dapat menggunakan fitur ini. Menjaga semua kredensial pengguna di dalam server LDAP dapat menghemat waktu dan upaya dengan menyediakan lokasi sentral untuk menyimpan serta mengelola kredensial ini.

Amazon MQ menyediakan dukungan LDAP menggunakan plugin Apache ActiveMQ JAAS. Setiap server LDAP, seperti Microsoft Active Directory atau OpenLDAP yang didukung oleh plugin juga didukung oleh Amazon MQ. Untuk informasi selengkapnya tentang plugin, lihat bagian Keamanan dalam dokumentasi ActiveMQ.

Selain pengguna, Anda dapat menentukan akses ke topik dan antrean untuk grup atau pengguna tertentu melalui server LDAP. Anda melakukannya dengan membuat entri yang mewakili topik dan antrean di server LDAP lalu menetapkan izin untuk pengguna atau grup LDAP tertentu. Kemudian Anda dapat mengonfigurasi broker untuk mengambil data otorisasi dari server LDAP.

> **Important**
>
> Saat menggunakan LDAP, otentikasi tidak peka huruf besar/kecil, tetapi otorisasi peka huruf besar/kecil untuk nama pengguna Anda.

Prasyarat

Sebelum menambahkan dukungan LDAP ke broker Amazon MQ baru atau yang sudah ada, Anda harus menyiapkan akun layanan. Akun layanan ini diperlukan untuk memulai koneksi ke server LDAP dan harus memiliki izin yang benar untuk membuat koneksi ini. Akun layanan ini akan menyiapkan autentikasi LDAP untuk broker Anda. Setiap koneksi klien berturut-turut akan diautentikasi melalui koneksi yang sama.

Akun layanan adalah akun di dalam server LDAP Anda, yang memiliki akses untuk memulai koneksi. Ini adalah persyaratan LDAP standar dan Anda harus memberikan kredensial akun layanan hanya satu kali. Setelah koneksi disiapkan, semua koneksi klien di masa mendatang akan dikonfirmasi melalui server LDAP Anda. Kredensial akun layanan Anda disimpan dengan aman dalam bentuk terenkripsi, yang hanya dapat diakses oleh Amazon MQ.

Untuk mengintegrasikan dengan ActiveMQ, Directory Information Tree (DIT) tertentu diperlukan dalam server LDAP. Misalnya, file `ldif` yang secara jelas menampilkan struktur ini, lihat Mengimpor file LDIF berikut ke server LDAP di bagian Keamanan dalam dokumentasi ActiveMQ.

Memulai dengan LDAP

Untuk memulai, arahkan ke konsol Amazon MQ dan pilih autentikasi dan otorisasi LDAP ketika Anda membuat Amazon MQ baru atau mengedit instans broker yang ada.

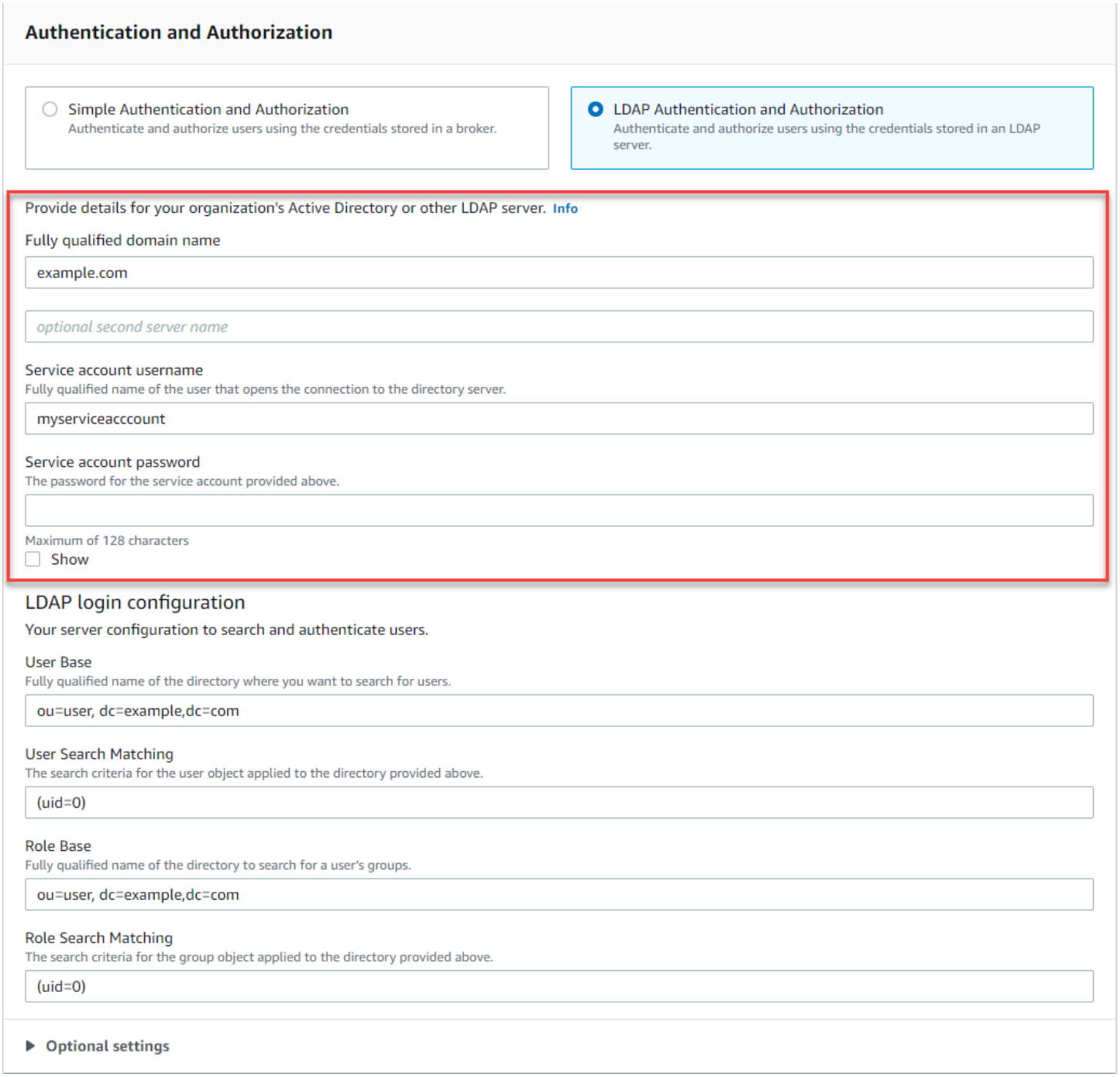
Berikan informasi berikut tentang akun layanan:

- Nama domain yang sepenuhnya memenuhi syarat Lokasi server LDAP tempat permintaan autentikasi dan otorisasi dikeluarkan.

> **Note**
>
> Nama domain yang sepenuhnya memenuhi syarat dari server LDAP yang harus Anda masukkan tidak boleh menyertakan protokol atau nomor port. Amazon MQ akan menambahkan nama domain yang sepenuhnya memenuhi syarat dengan protokol `ldaps`, dan akan menambahkan nomor port `636`.
>
> Sebagai contoh, jika Anda memberikan domain yang sepenuhnya memenuhi syarat berikut: `example.com`, Amazon MQ akan mengakses server LDAP menggunakan URL berikut: `ldaps://example.com:636`.
>
> Agar host broker dapat berhasil berkomunikasi dengan server LDAP, nama domain yang sepenuhnya memenuhi syarat harus dapat dibuat secara publik. Untuk menjaga server LDAP tetap privat dan aman, batasi lalu lintas masuk dalam aturan masuk server untuk hanya mengizinkan lalu lintas yang berasal dari dalam VPC broker.

- Nama pengguna akun layanan Nama pengguna yang khas dan akan digunakan untuk melakukan ikatan awal ke server LDAP.
- Kata sandi akun layanan Kata sandi pengguna yang melakukan ikatan awal.

Gambar berikut menyoroti tempat untuk memasukkan detail ini.

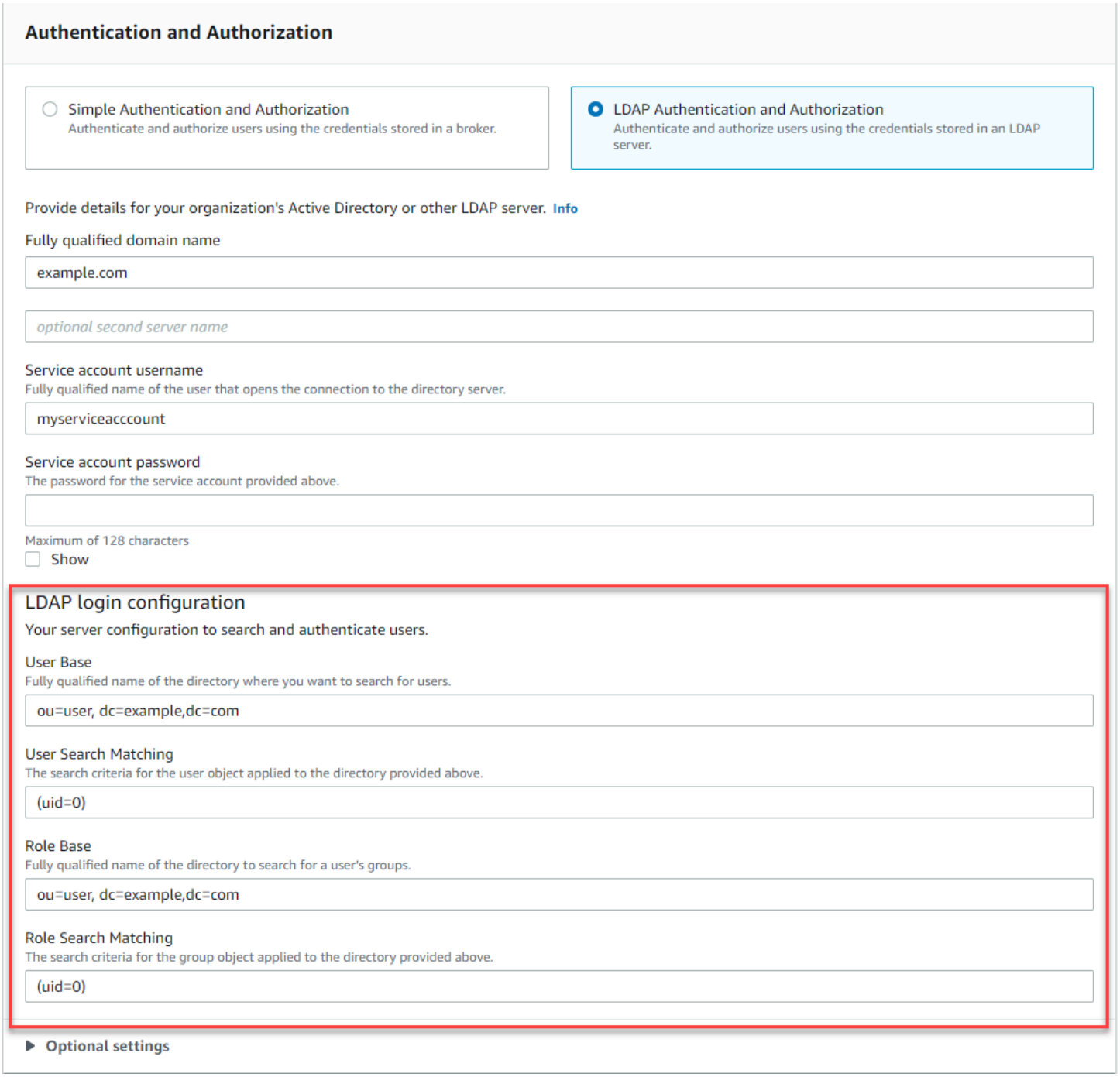


Di bagian Konfigurasi login LDAP, berikan informasi yang diperlukan berikut:

- Basis Pengguna Nama simpul yang khas dalam directory information tree (DIT) dan merupakan tempat pencarian pengguna.
- Pencocokan Pencarian Pengguna Filter pencarian LDAP yang akan digunakan untuk menemukan pengguna dalam `userBase`. Nama pengguna klien akan diganti ke dalam placeholder `{0}` di filter pencarian. Untuk informasi lebih lanjut, lihat Autentikasi dan Otorisasi.

- Basis Peran Nama simpul yang khas dalam DIT dan merupakan tempat pencarian peran. Peran dapat dikonfigurasi sebagai entri grup LDAP eksplisit dalam direktori Anda. Entri peran umum dapat terdiri dari satu atribut untuk nama peran, seperti nama umum (CN), dan atribut lain, seperti `member`, dengan nilai yang mewakili nama khas atau nama pengguna yang termasuk dalam grup peran. Sebagai contoh, dengan unit organisasi, `group`, Anda dapat memberikan nama khas berikut: `ou=group,dc=example,dc=com`.
- Pencocokan Pencarian Peran Filter pencarian LDAP yang akan digunakan untuk menemukan peran dalam `roleBase`. Nama khas pengguna dicocokkan yang menurut `userSearchMatching` akan diganti ke dalam placeholder `{0}` di filter pencarian. Nama pengguna klien akan diganti dalam placeholder `{1}`. Misalnya, jika entri peran dalam direktori Anda menyertakan atribut bernama `member`, yang berisi nama pengguna untuk semua pengguna dalam peran tersebut, Anda dapat menyediakan filter pencarian berikut: `(member:=uid={1})`.

Gambar berikut menyoroti tempat untuk menentukan detail ini.



Di bagian Pengaturan opsional, Anda dapat memberikan informasi opsional berikut:

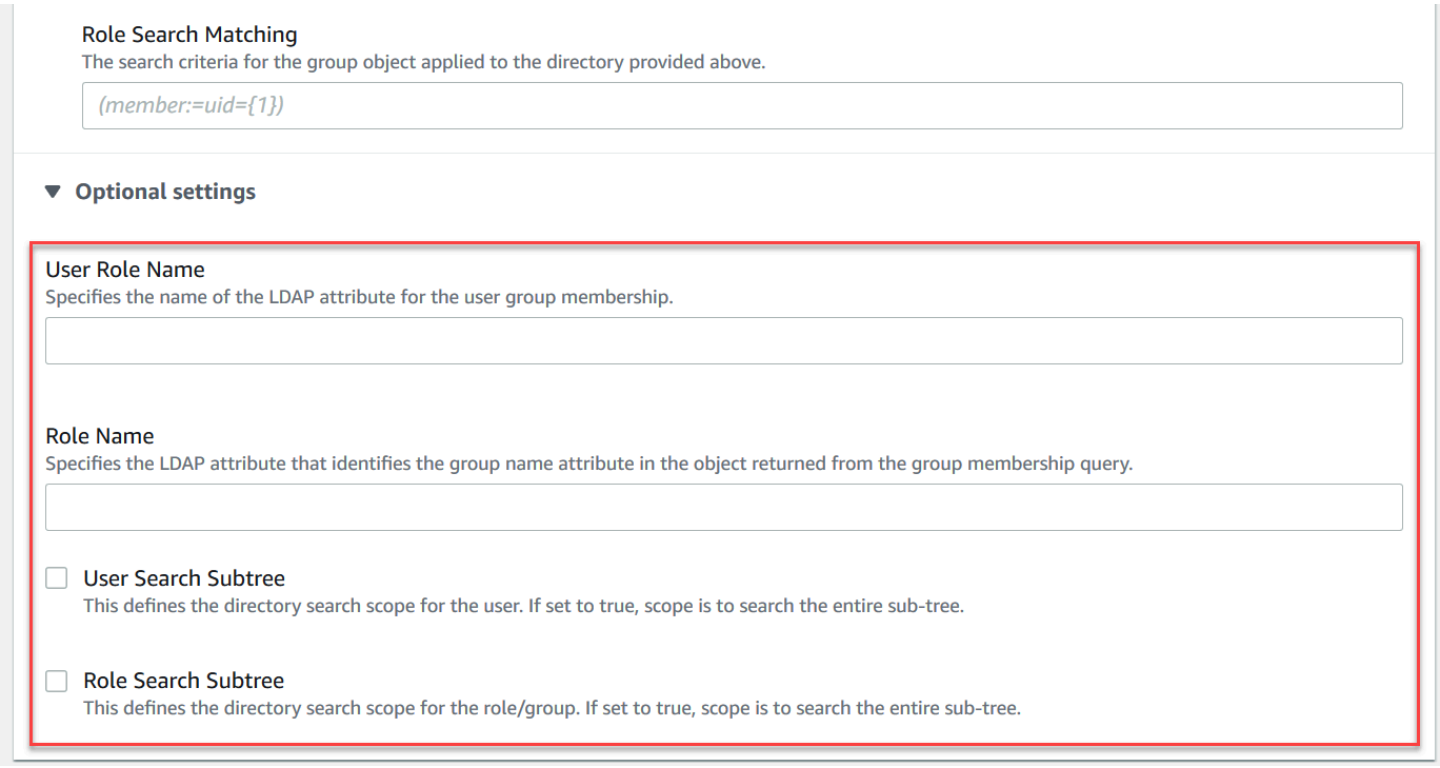
- Nama Peran Pengguna Nama atribut LDAP dalam entri direktori pengguna untuk keanggotaan grup pengguna. Dalam beberapa kasus, peran pengguna dapat diidentifikasi menurut nilai atribut dalam entri direktori pengguna. Opsi `userRoleName` memungkinkan Anda untuk memberikan nama bagi atribut ini. Sebagai contoh, mari kita pertimbangkan entri pengguna berikut:

```
dn: uid=jdoe,ou=user,dc=example,dc=com
objectClass: user
uid: jdoe
sn: jane
cn: Jane Doe
mail: j.doe@somecompany.com
memberOf: role1
userPassword: password
```

Untuk memberikan`userRoleName` yang benar bagi contoh di atas, Anda akan menentukan atribut `memberOf`. Jika autentikasi berhasil, pengguna ditetapkan peran `role1`.

- Nama Peran Atribut nama grup dalam entri peran yang nilainya adalah nama peran tersebut. Misalnya, Anda dapat menentukan `cn` untuk entri grup nama umum. Jika autentikasi berhasil, pengguna ditetapkan nilai atribut `cn` untuk setiap entri peran tempat mereka menjadi anggota.
- Subpohon Pencarian Pengguna Menentukan ruang lingkup untuk kueri pencarian pengguna LDAP. Jika benar, ruang lingkup diatur untuk mencari seluruh subpohon di bawah simpul yang ditentukan menurut `userBase`.
- Subpohon Pencarian Peran Menentukan ruang lingkup untuk kueri pencarian peran LDAP. Jika benar, ruang lingkup diatur untuk mencari seluruh subpohon di bawah simpul yang ditentukan menurut `roleBase`.

Gambar berikut menyoroti tempat untuk menentukan pengaturan opsional ini.



Cara kerja integrasi LDAP

Anda bisa memikirkan integrasi dalam dua kategori utama: struktur untuk autentikasi, dan struktur untuk otorisasi.

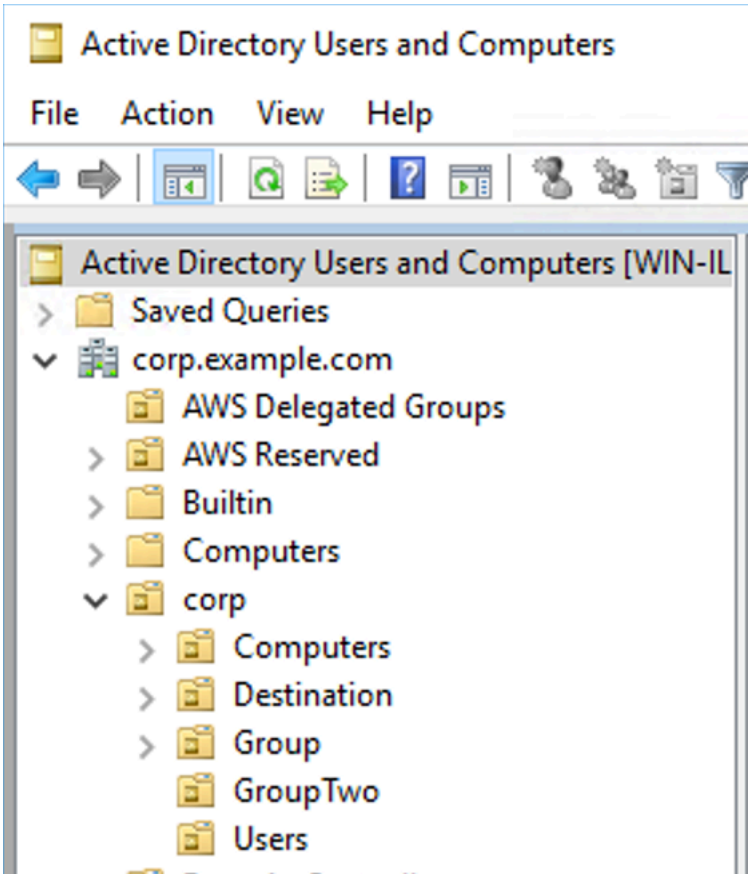
Autentikasi

Untuk autentikasi, kredensial klien harus valid. Kredensial ini divalidasi terhadap pengguna di basis pengguna dalam server LDAP.

Basis pengguna yang disediakan untuk broker ActiveMQ harus menunjuk ke simpul di DIT tempat pengguna disimpan dalam server LDAP. Misalnya, jika Anda menggunakan AWS Managed Microsoft AD, dan Anda memiliki komponen domain,, dan `corp` `examplecom`, dan di dalamnya Anda memiliki unit organisasi `corp` dan`Users`, Anda akan menggunakan yang berikut ini sebagai basis pengguna Anda:

```
OU=Users,OU=corp,DC=corp,DC=example,DC=com
```

Broker ActiveMQ akan mencari pengguna di lokasi ini dalam DIT guna mengautentikasi permintaan koneksi klien ke broker.



Karena kode sumber ActiveMQ meng-hardcode nama atribut untuk pengguna menjadi `uid`, Anda harus memastikan bahwa setiap pengguna telah menetapkan atribut ini. Untuk lebih sederhana, Anda dapat menggunakan nama pengguna koneksi pengguna. Untuk informasi selengkapnya, lihat kode sumber activemq dan Mengonfigurasi pemetaan ID di Pengguna dan Komputer Direktori Aktif untuk versi Windows Server 2016 (dan berikutnya).

Untuk mengaktifkan akses konsol ActiveMQ bagi pengguna tertentu, pastikan mereka merupakan anggota grup `amazonmq-console-admins`.

Otorisasi

Untuk otorisasi, basis pencarian izin ditentukan dalam konfigurasi broker. Otorisasi dilakukan dengan basis per tujuan (atau wildcard, set tujuan) melalui elemen `cachedLdapAuthorizationMap`, yang ditemukan dalam file konfigurasi `activemq.xml` broker. Untuk informasi selengkapnya, lihat Modul Otorisasi LDAP yang Di-cache.

> **Note**
>
> Untuk dapat menggunakan `cachedLDAPAuthorizationMap` elemen dalam file `activemq.xml` konfigurasi broker Anda, Anda harus memilih opsi Otentikasi dan Otorisasi LDAP saat membuat konfigurasi melalui Konsol Manajemen AWS, atau mengatur pembuatan konfigurasi melalui Konsol Manajemen AWS, atau mengatur `authenticationStrategy`properti LDAP saat membuat konfigurasi baru menggunakan Amazon MQ API.

Anda harus memberikan tiga atribut berikut sebagai bagian dari elemen `cachedLDAPAuthorizationMap`:

- `queueSearchBase`
- `topicSearchBase`
- `tempSearchBase`

> **Important**
>
> Agar informasi sensitif tidak langsung ditempatkan ke file konfigurasi broker, Amazon MQ memblokir atribut berikut dari agar tidak digunakan dalam `cachedLdapAuthorizationMap`:
>
> - `connectionURL`
> - `connectionUsername`
> - `connectionPassword`
>
> Saat Anda membuat broker, Amazon MQ mengganti nilai yang Anda berikan melalui Konsol Manajemen AWS, atau di `ldapServerMetadata`properti permintaan API Anda, untuk atribut di atas.

Hal berikut mendemonstrasikan contoh kerja `cachedLdapAuthorizationMap`.

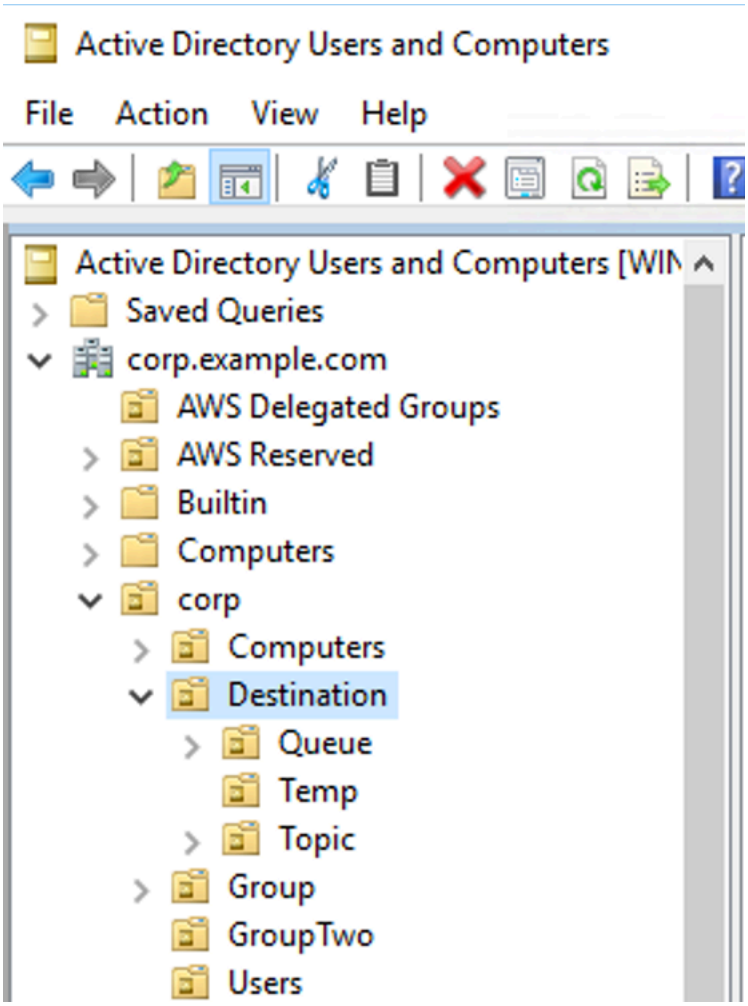
```
<authorizationPlugin>
    <map>
```

```
        <cachedLDAPAuthorizationMap
            queueSearchBase="ou=Queue,ou=Destination,ou=corp,dc=corp,dc=example,dc=com"
            topicSearchBase="ou=Topic,ou=Destination,ou=corp,dc=corp,dc=example,dc=com"
            tempSearchBase="ou=Temp,ou=Destination,ou=corp,dc=corp,dc=example,dc=com"
            refreshInterval="300000"
            legacyGroupMapping="false"
        />
    </map>
</authorizationPlugin>
```

Nilai ini mengidentifikasi lokasi dalam DIT tempat izin untuk setiap jenis tujuan ditentukan. Jadi untuk contoh di atas dengan AWS Managed Microsoft AD, menggunakan komponen domain yang sama dari `corpexample`,`com`, dan, Anda akan menentukan unit organisasi bernama `destination` berisi semua jenis tujuan Anda. Dalam OU tersebut, Anda akan membuat satu untuk `queues`, satu untuk `topics`, dan satu untuk tujuan `temp`.

Ini berarti basis pencarian antrean Anda, yang menyediakan informasi otorisasi untuk tujuan antrean jenis, akan memiliki lokasi berikut di DIT Anda:

```
OU=Queue,OU=Destination,OU=corp,DC=corp,DC=example,DC=com
```



Demikian pula, aturan izin untuk topik dan tujuan sementara akan terletak pada tingkat yang sama di DIT:

```
OU=Topic,OU=Destination,OU=corp,DC=corp,DC=example,DC=com
OU=Temp,OU=Destination,OU=corp,DC=corp,DC=example,DC=com
```

Dalam OU untuk setiap jenis tujuan (antrean, topik, sementara), baik wildcard atau nama tujuan tertentu dapat disediakan. Misalnya, untuk memberikan aturan otorisasi bagi semua antrean yang dimulai dengan prefiks DEMO.EVENTS.$., Anda dapat membuat OU berikut:

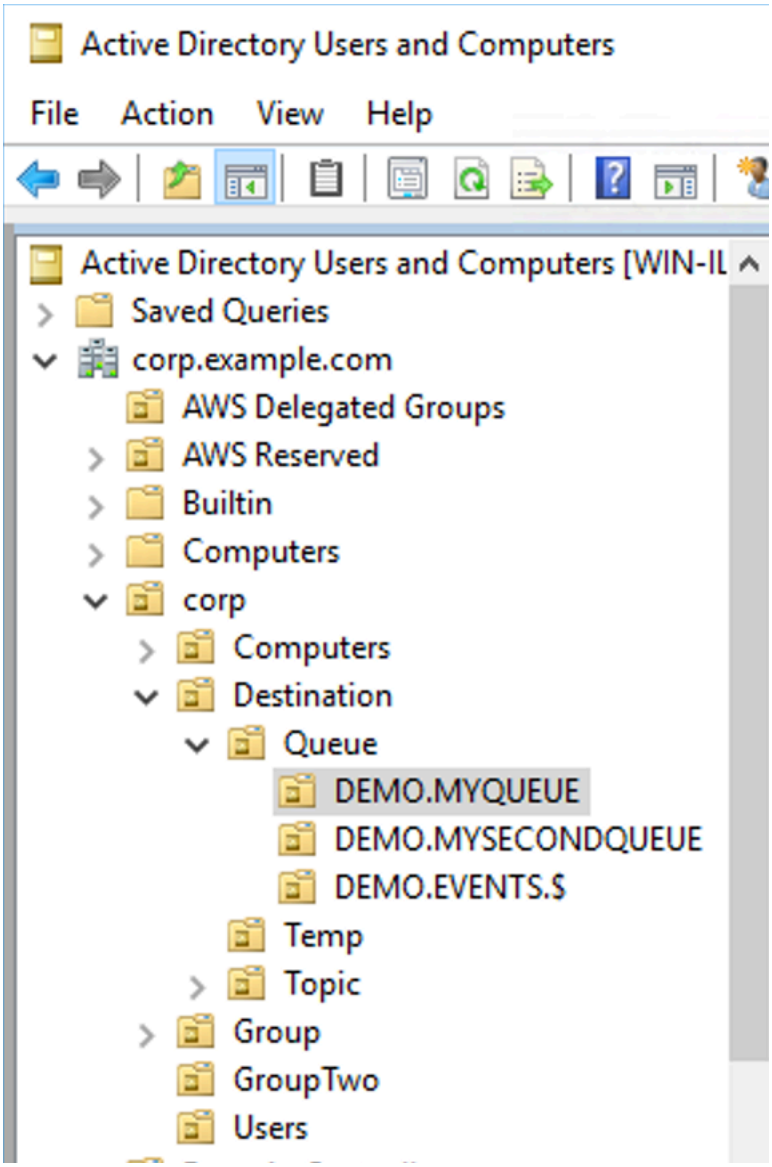
```
OU=DEMO.EVENTS.$,OU=Queue,OU=Destination,OU=corp,DC=corp,DC=example,DC=com
```

> **Note**
>
> OU `DEMO.EVENTS.$` berada di dalam OU `Queue`.

Untuk info selengkapnya tentang wildcard di ActiveMQ, lihat Wildcard

Untuk memberikan aturan otorisasi bagi antrean tertentu, seperti DEMO.MYQUEUE, tentukan hal seperti berikut:

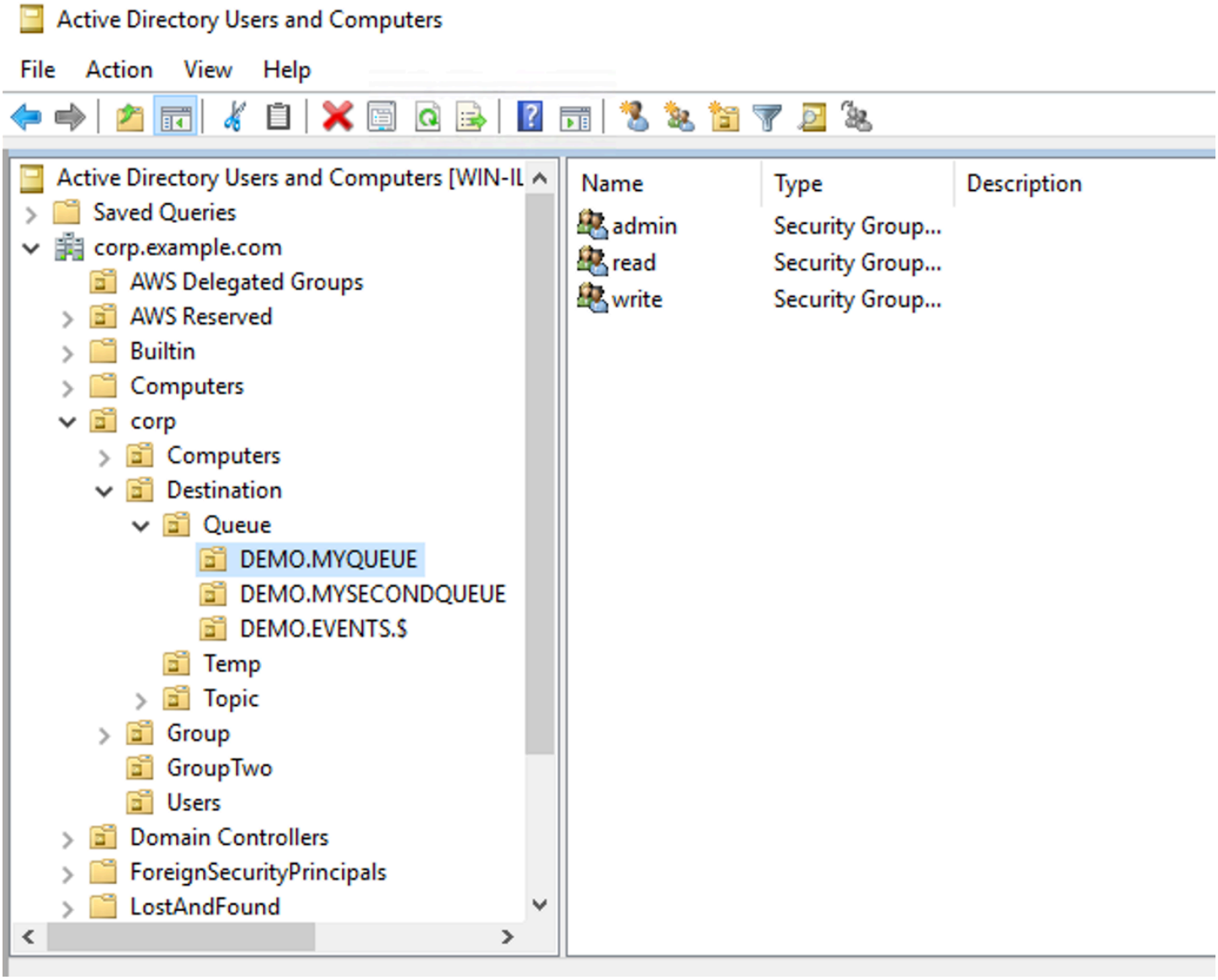
```
OU=DEMO.MYQUEUE,OU=Queue,OU=Destination,OU=corp,DC=corp,DC=example,DC=com
```



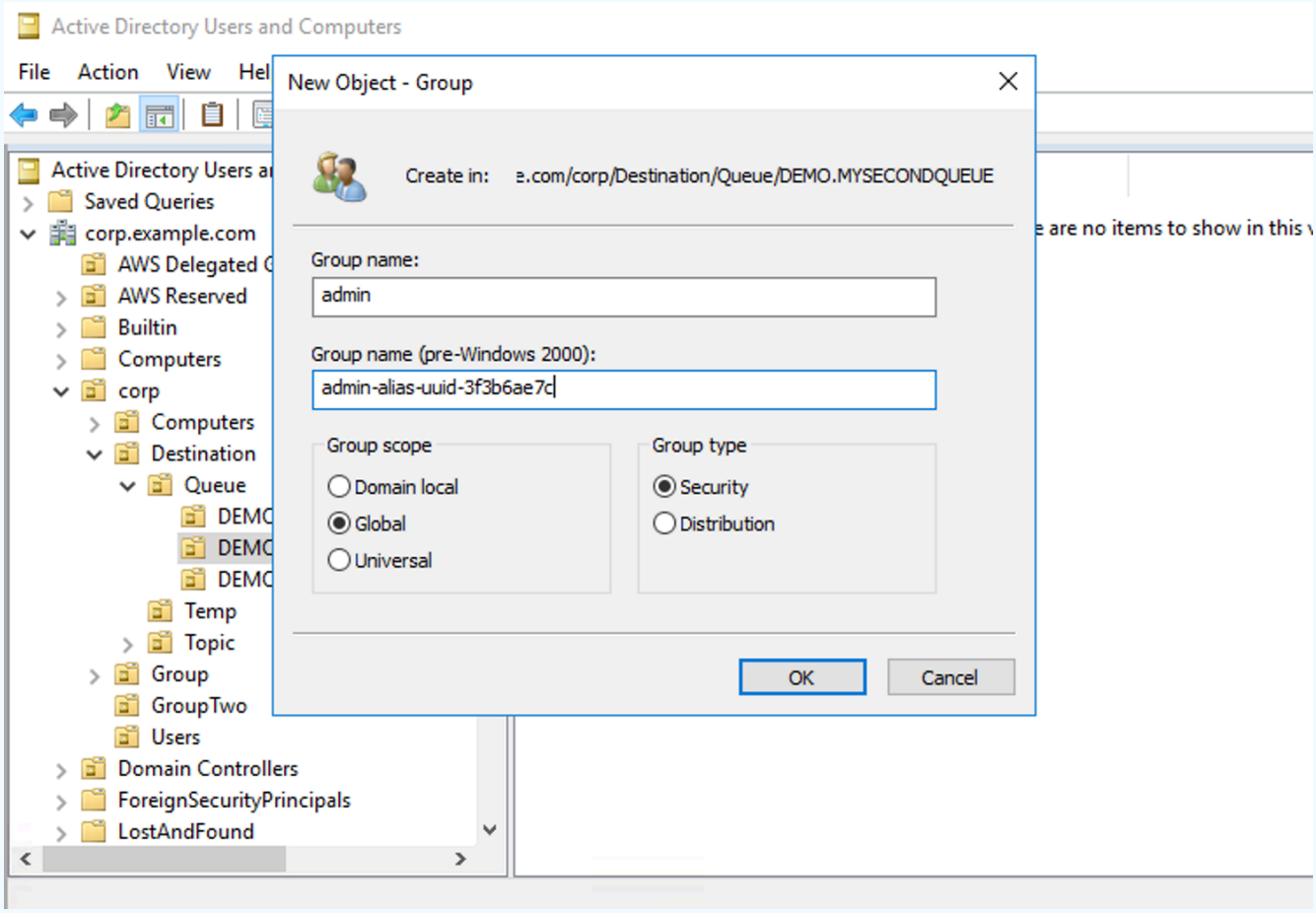
Grup Keamanan

Dalam setiap OU yang mewakili tujuan atau wildcard, Anda harus membuat tiga grup keamanan. Seperti semua izin di ActiveMQ, ini adalah izin. read/write/admin Untuk informasi selengkapnya tentang hal yang dapat dilakukan pengguna dengan setiap izin tersebut, lihat Keamanan dalam dokumentasi ActiveMQ.

Anda harus memberi nama grup keamanan ini `read`, `write`, dan `admin`. Dalam setiap grup keamanan ini, Anda dapat menambahkan pengguna atau grup, yang kemudian akan memiliki izin untuk melakukan tindakan terkait. Anda memerlukan grup keamanan ini untuk setiap rangkaian tujuan wildcard atau tujuan individual.



> **Note**
>
> Ketika Anda membuat grup admin, konflik akan muncul dengan nama grup. Konflik ini terjadi karena aturan warisan pra-Windows 2000 tidak mengizinkan grup untuk berbagi nama yang sama, bahkan jika grup berada di lokasi DIT yang berbeda. Nilai di dalam kotak teks pra-Windows 2000 tidak berdampak pada penyiapan, tetapi harus unik secara global. Untuk menghindari konflik ini, Anda dapat menambahkan sufiks `uuid` ke setiap grup `admin`.



Menambahkan pengguna ke grup keamanan `admin` untuk tujuan tertentu akan memungkinkan pengguna untuk membuat dan menghapus topik tersebut. Menambahkannya ke grup keamanan `read` akan memungkinkan mereka untuk membaca dari tujuan, dan menambahkannya ke grup `write` akan memungkinkan mereka untuk menulis ke tujuan.

Selain menambahkan pengguna individu ke izin grup keamanan, Anda juga dapat menambahkan seluruh grup. Namun, karena ActiveMQ meng-hardcode atribut nama untuk grup, Anda harus

memastikan bahwa grup yang ingin Anda tambahkan memiliki kelas objek `groupOfNames`, seperti yang ditampilkan dalam kode sumber activemq.

Untuk melakukannya, ikuti proses yang seperti `uid` bagi pengguna. Lihat Mengonfigurasi pemetaan ID di Pengguna dan Komputer Direktori Aktif untuk versi Windows Server 2016 (dan berikutnya).

Langkah 3: (Opsional) Connect ke AWS Lambda fungsi

AWS Lambda dapat terhubung ke dan mengkonsumsi pesan dari broker Amazon MQ Anda. Saat Anda menghubungkan broker ke Lambda, Anda membuat pemetaan sumber peristiwa yang membaca pesan dari antrian dan memanggil fungsi secara sinkron. Pemetaan sumber acara yang Anda buat membaca pesan dari broker Anda dalam batch dan mengubahnya menjadi muatan Lambda dalam bentuk objek JSON.

Untuk menghubungkan broker Anda ke fungsi Lambda

1. Tambahkan izin peran IAM berikut ke peran eksekusi fungsi Lambda Anda.

   - mq: DescribeBroker
   - EC2: CreateNetworkInterface
   - EC2: DeleteNetworkInterface
   - EC2: DescribeNetworkInterfaces
   - EC2: DescribeSecurityGroups
   - EC2: DescribeSubnets
   - EC2: DescribeVpcs
   - log: CreateLogGroup
   - log: CreateLogStream
   - log: PutLogEvents
   - manajer rahasia: GetSecretValue

   > **Note**
   >
   > Tanpa izin IAM yang diperlukan, fungsi Anda tidak akan berhasil membaca catatan dari sumber daya Amazon MQ.

2. (Opsional) Jika Anda telah membuat broker tanpa aksesibilitas publik, Anda harus melakukan salah satu hal berikut untuk memungkinkan Lambda terhubung ke broker Anda:

   - Konfigurasikan satu NAT gateway per subnet publik. Untuk informasi selengkapnya, lihat Akses Internet dan layanan untuk fungsi yang terhubung dengan VPC di Panduan Pengembang.AWS Lambda
   - Buat koneksi antara Amazon Virtual Private Cloud (Amazon VPC) dan Lambda menggunakan titik akhir VPC. VPC Amazon Anda juga harus terhubung ke AWS Security Token Service (AWS STS) dan titik akhir Secrets Manager. Untuk informasi selengkapnya, lihat Mengonfigurasi titik akhir VPC antarmuka untukAWS Lambda Lambda di Panduan Pengembang.

3. Konfigurasikan broker Anda sebagai sumber acara untuk fungsi Lambda menggunakan. Konsol Manajemen AWS Anda juga dapat menggunakan `create-event-source-mapping` AWS Command Line Interface perintah.

4. Tulis beberapa kode untuk fungsi Lambda Anda untuk memproses pesan yang dikonsumsi dari broker Anda. Payload Lambda yang diambil oleh pemetaan sumber peristiwa Anda tergantung pada jenis mesin broker. Berikut ini adalah contoh payload Lambda untuk Amazon MQ untuk antrian ActiveMQ.

   > **Note**
   >
   > Dalam contoh, `testQueue` adalah nama antrian.

```
{
  "eventSource": "aws:amq",
  "eventSourceArn": "arn:aws:mq:us-
west-2:112556298976:broker:test:b-9bcfa592-423a-4942-879d-eb284b418fc8",
  "messages": {
    [
      {
        "messageID": "ID:b-9bcfa592-423a-4942-879d-eb284b418fc8-1.mq.us-
west-2.amazonaws.com-37557-1234520418293-4:1:1:1:1",
        "messageType": "jms/text-message",
        "data": "QUJDOkFBQUE=",
        "connectionId": "myJMSCoID",
        "redelivered": false,
        "destination": {
          "physicalname": "testQueue"
```

```
        },
        "timestamp": 1598827811958,
        "brokerInTime": 1598827811958,
        "brokerOutTime": 1598827811959
      },
      {
        "messageID": "ID:b-9bcfa592-423a-4942-879d-eb284b418fc8-1.mq.us-
west-2.amazonaws.com-37557-1234520418293-4:1:1:1:1",
        "messageType":"jms/bytes-message",
        "data": "3DTOOW7crj51prgVLQaGQ82S48k=",
        "connectionId": "myJMSCoID1",
        "persistent": false,
        "destination": {
          "physicalname": "testQueue"
        },
        "timestamp": 1598827811958,
        "brokerInTime": 1598827811958,
        "brokerOutTime": 1598827811959
      }
    ]
  }
}
```

Untuk informasi selengkapnya tentang menghubungkan Amazon MQ ke Lambda, opsi yang didukung Lambda untuk sumber peristiwa Amazon MQ, dan kesalahan pemetaan sumber peristiwa, lihat Menggunakan Lambda dengan Amazon MQ di Panduan Pengembang.AWS Lambda

Membuat pengguna broker ActiveMQ

Pengguna ActiveMQ adalah orang atau aplikasi yang dapat mengakses antrean dan topik broker ActiveMQ. Anda dapat mengonfigurasi pengguna untuk memiliki izin tertentu. Misalnya, Anda dapat mengizinkan beberapa pengguna mengakses Konsol Web ActiveMQ.

Grup adalah label semantik. Anda dapat menetapkan grup ke pengguna dan mengonfigurasi izin untuk grup untuk mengikirim ke, menerima dari, dan mengelola antrean serta topik tertentu.

> **Note**
>
> Anda tidak dapat mengonfigurasi grup pengguna secara independen. Label grup dibuat saat Anda menambahkan setidaknya satu pengguna ke grup dan dihapus saat Anda menghapus semua pengguna dari grup.

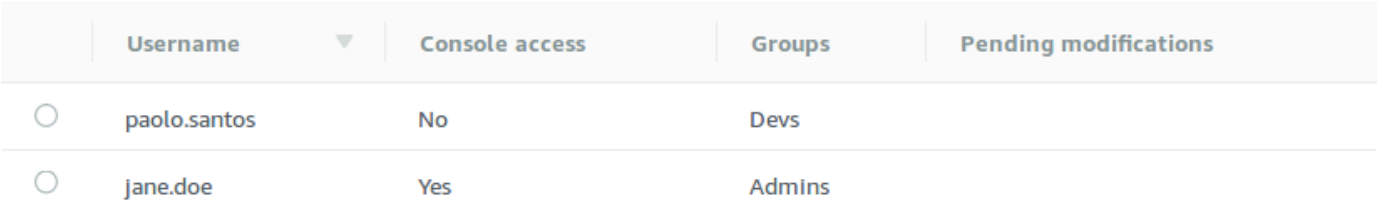
> **Note**
>
> `activemq-webconsole`Grup di ActiveMQ di Amazon MQ memiliki izin admin pada semua antrian dan topik. Semua pengguna dalam grup ini akan memiliki akses admin.

Contoh berikut menunjukkan cara membuat, mengedit, dan menghapus pengguna broker Amazon MQ menggunakan Konsol Manajemen AWS.

Buat pengguna broker ActiveMQ baru

1. Masuk ke konsol Amazon MQ.
2. Dari daftar broker, pilih nama broker Anda (misalnya, MyBroker) dan kemudian pilih Lihat detail.

   Pada ***MyBroker***halaman, di bagian Pengguna, semua pengguna untuk broker ini terdaftar.

3. Pilih Buat pengguna.
4. Di kotak dialog Buat pengguna, ketik Nama pengguna dan Kata sandi.
5. (Opsional) Ketik nama grup tempat pengguna berada, dipisahkan dengan koma (misalnya: `Devs, Admins`).
6. (Opsional) Untuk mengizinkan pengguna mengakses Konsol Web ActiveMQ, pilih Konsol Web ActiveMQ.
7. Pilih Buat pengguna.

> **Important**
>
> Pembuatan perubahan pada pengguna tidak akan segera menerapkan perubahan ke pengguna. Untuk menerapkan perubahan Anda, Anda harus menunggu jendela pemeliharaan berikutnya atau reboot broker.

Edit pengguna broker ActiveMQ

Untuk mengedit pengguna yang sudah ada, lakukan hal berikut:

1. Masuk ke konsol Amazon MQ.
2. Dari daftar broker, pilih nama broker Anda (misalnya, MyBroker) dan kemudian pilih Lihat detail.

   Pada ***MyBroker***halaman, di bagian Pengguna, semua pengguna untuk broker ini terdaftar.

	Username	Console access	Groups	Pending modifications
○	paolo.santos	No	Devs	
○	jane.doe	Yes	Admins	

3. Pilih kredenal masuk Anda dan pilih Edit.

   Kotak dialog Edit pengguna akan ditampilkan.
4. (Opsional) Ketik Kata Sandi baru.
5. (Opsional) Tambahkan atau hapus nama grup tempat pengguna berada, dipisahkan dengan koma (misalnya: `Managers, Admins`).
6. (Opsional) Untuk mengizinkan pengguna mengakses Konsol Web ActiveMQ, pilih Konsol Web ActiveMQ.
7. Untuk menyimpan perubahan pada pengguna, pilih Selesai.

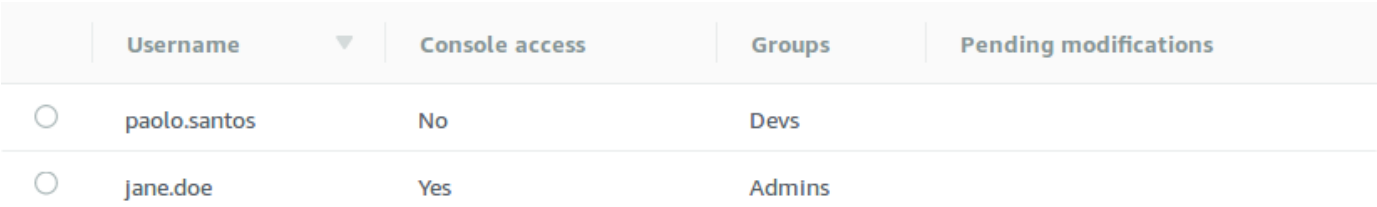
> **Important**
>
> Pembuatan perubahan pada pengguna tidak akan segera menerapkan perubahan ke pengguna. Untuk menerapkan perubahan Anda, Anda harus menunggu jendela pemeliharaan berikutnya atau reboot broker.

Hapus pengguna broker ActiveMQ

Ketika Anda tidak lagi membutuhkan pengguna, Anda dapat menghapus pengguna.

1. Masuk ke konsol Amazon MQ.
2. Dari daftar broker, pilih nama broker Anda (misalnya, MyBroker) dan kemudian pilih Lihat detail.

   Pada ***MyBroker***halaman, di bagian Pengguna, semua pengguna untuk broker ini terdaftar.



	Username	Console access	Groups	Pending modifications
○	paolo.santos	No	Devs	
○	jane.doe	Yes	Admins	

3. Pilih kredensi login Anda (misalnya, ***MyUser***) lalu pilih Hapus.
4. Untuk mengonfirmasi penghapusan pengguna, di Hapus? ***MyUser*** kotak dialog, pilih Hapus.

> **Important**
>
> Pembuatan perubahan pada pengguna tidak akan segera menerapkan perubahan ke pengguna. Untuk menerapkan perubahan Anda, Anda harus menunggu jendela pemeliharaan berikutnya atau reboot broker.

Contoh kerja menggunakan Java Message Service (JMS) dengan ActiveMQ

Contoh berikut menunjukkan cara bekerja dengan ActiveMQ secara pemrograman:

- OpenWire Contoh kode Java terhubung ke broker, membuat antrian, dan mengirim dan menerima pesan. Untuk rincian dan penjelasan detail, lihat Connecting a Java application to your broker.
- Kode Java contoh MQTT terhubung ke broker, membuat topik, serta memublikasikan dan menerima pesan.
- Kode Java contoh STOMP+WSS terhubung ke broker, membuat antrean, serta memublikasikan dan menerima pesan.

Prasyarat

Mengaktifkan Atribut VPC

Untuk memastikan bahwa broker dapat diakses dalam VPC, Anda harus mengaktifkan atribut VPC `enableDnsHostnames` dan `enableDnsSupport`. Untuk informasi selengkapnya, lihat Dukungan DNS di VPC Anda dalam Panduan Pengguna Amazon VPC.

Mengaktifkan Koneksi masuk

Untuk bekerja dengan Amazon MQ secara terprogram, Anda harus menggunakan koneksi masuk.

1. Masuk ke konsol Amazon MQ.
2. Dari daftar broker, pilih nama broker Anda (misalnya, MyBroker).
3. Pada ***`MyBroker`***halaman, di bagian Koneksi, catat alamat dan port URL konsol web broker dan protokol tingkat kabel.
4. Di bagian Detail, di bawah Keamanan dan jaringan, pilih nama grup keamanan Anda atau
   .

   Halaman Grup Keamanan Dasbor EC2 akan ditampilkan.
5. Dari daftar grup keamanan, pilih grup keamanan Anda.
6. Di bagian bawah halaman, pilih tab Masuk, lalu pilih Edit.
7. Di kotak dialog Edit aturan masuk, tambahkan aturan untuk setiap URL atau titik akhir yang Anda inginkan untuk dapat diakses secara publik (contoh berikut menampilkan cara melakukannya untuk konsol web broker).
   a. Pilih Tambahkan aturan.
   b. Untuk Jenis, pilih TCP Kustom.
   c. Untuk Rentang Port, ketik port konsol web (`8162`).
   d. Untuk Sumber, biarkan Kustom dipilih lalu ketik alamat IP sistem yang Anda inginkan untuk dapat mengakses konsol web (misalnya, `192.0.2.1`).
   e. Pilih Simpan.

      Broker Anda kini dapat menerima koneksi masuk.

Menambahkan dependensi Java

OpenWire

Tambahkan paket `activemq-client.jar` dan `activemq-pool.jar` ke jalur kelas Java Anda. Contoh berikut menampilkan dependensi ini dalam file `pom.xml` proyek Maven.

```
<dependencies>
    <dependency>
        <groupId>org.apache.activemq</groupId>
        <artifactId>activemq-client</artifactId>
        <version>5.15.16</version>
    </dependency>
    <dependency>
        <groupId>org.apache.activemq</groupId>
        <artifactId>activemq-pool</artifactId>
        <version>5.15.16</version>
    </dependency>
</dependencies>
```

Untuk informasi selengkapnya tentang `activemq-client.jar`, lihat Konfigurasi Awal dalam dokumentasi Apache ActiveMQ.

MQTT

Tambahkan paket `org.eclipse.paho.client.mqttv3.jar` ke jalur kelas Java Anda. Contoh berikut menampilkan dependensi ini dalam file `pom.xml` proyek Maven.

```
<dependencies>
                    <dependency>
                        <groupId>org.eclipse.paho</groupId>
                        <artifactId>org.eclipse.paho.client.mqttv3</artifactId>
                        <version>1.2.0</version>
                    </dependency>
                    </dependencies>
```

Untuk informasi selengkapnya tentang `org.eclipse.paho.client.mqttv3.jar`, lihat Klien Java Eclipse Paho.

STOMP+WSS

Tambahkan paket berikut ke jalur kelas Java Anda:

- spring-messaging.jar
- spring-websocket.jar
- javax.websocket-api.jar
- jetty-all.jar
- slf4j-simple.jar
- jackson-databind.jar

Contoh berikut menampilkan dependensi ini dalam file pom.xml proyek Maven.

```
<dependencies>
                    <dependency>
                        <groupId>org.springframework</groupId>
                        <artifactId>spring-messaging</artifactId>
                        <version>5.0.5.RELEASE</version>
                    </dependency>
                    <dependency>
                        <groupId>org.springframework</groupId>
                        <artifactId>spring-websocket</artifactId>
                        <version>5.0.5.RELEASE</version>
                    </dependency>
                    <dependency>
                        <groupId>javax.websocket</groupId>
                        <artifactId>javax.websocket-api</artifactId>
                        <version>1.1</version>
                    </dependency>
                    <dependency>
                        <groupId>org.eclipse.jetty.aggregate</groupId>
                        <artifactId>jetty-all</artifactId>
                        <type>pom</type>
                        <version>9.3.3.v20150827</version>
                    </dependency>
                    <dependency>
                        <groupId>org.slf4j</groupId>
                        <artifactId>slf4j-simple</artifactId>
                        <version>1.6.6</version>
                    </dependency>
                    <dependency>
                        <groupId>com.fasterxml.jackson.core</groupId>
                        <artifactId>jackson-databind</artifactId>
                        <version>2.5.0</version>
```

```
                    </dependency>
                    </dependencies>
```

Untuk informasi selengkapnya, lihat Dukungan STOMP dalam dokumentasi Spring Framework.

Amazon MQExample .java

> **Important**
>
> Pada kode contoh berikut, produsen dan konsumen berjalan dalam satu utas. Untuk sistem produksi (atau untuk menguji failover instans broker), pastikan bahwa produsen dan konsumen berjalan di host atau utas terpisah.

OpenWire

```
/*
 * Copyright 2010-2019 Amazon.com, Inc. or its affiliates. All Rights Reserved.
 *
 * Licensed under the Apache License, Version 2.0 (the "License").
 * You may not use this file except in compliance with the License.
 * A copy of the License is located at
 *
 *  https://aws.amazon.com/apache2.0
 *
 * or in the "license" file accompanying this file. This file is distributed
 * on an "AS IS" BASIS, WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either
 * express or implied. See the License for the specific language governing
 * permissions and limitations under the License.
 *
 */

                    import org.apache.activemq.ActiveMQConnectionFactory;
                    import org.apache.activemq.jms.pool.PooledConnectionFactory;

                    import javax.jms.*;

                    public class AmazonMQExample {

                    // Specify the connection parameters.
                    private final static String WIRE_LEVEL_ENDPOINT
```

```
                            = "ssl://b-1234a5b6-78cd-901e-2fgh-3i45j6k178l9-1.mq.us-
east-2.amazonaws.com:61617";
                    private final static String ACTIVE_MQ_USERNAME =
 "MyUsername123";
                    private final static String ACTIVE_MQ_PASSWORD =
 "MyPassword456";

                    public static void main(String[] args) throws JMSException {
                        final ActiveMQConnectionFactory connectionFactory =
                                createActiveMQConnectionFactory();
                        final PooledConnectionFactory pooledConnectionFactory =
                                createPooledConnectionFactory(connectionFactory);

                        sendMessage(pooledConnectionFactory);
                        receiveMessage(connectionFactory);

                        pooledConnectionFactory.stop();
                    }

                    private static void
                    sendMessage(PooledConnectionFactory pooledConnectionFactory)
 throws JMSException {
                        // Establish a connection for the producer.
                        final Connection producerConnection =
 pooledConnectionFactory
                                .createConnection();
                        producerConnection.start();

                        // Create a session.
                        final Session producerSession = producerConnection
                                .createSession(false, Session.AUTO_ACKNOWLEDGE);

                        // Create a queue named "MyQueue".
                        final Destination producerDestination = producerSession
                                .createQueue("MyQueue");

                        // Create a producer from the session to the queue.
                        final MessageProducer producer = producerSession
                                .createProducer(producerDestination);
                        producer.setDeliveryMode(DeliveryMode.NON_PERSISTENT);

                        // Create a message.
                        final String text = "Hello from Amazon MQ!";
                        final TextMessage producerMessage = producerSession
```

```
                                .createTextMessage(text);

                        // Send the message.
                        producer.send(producerMessage);
                        System.out.println("Message sent.");

                        // Clean up the producer.
                        producer.close();
                        producerSession.close();
                        producerConnection.close();
                    }

                    private static void
                    receiveMessage(ActiveMQConnectionFactory connectionFactory)
throws JMSException {
                        // Establish a connection for the consumer.
                        // Note: Consumers should not use PooledConnectionFactory.
                        final Connection consumerConnection =
connectionFactory.createConnection();
                        consumerConnection.start();

                        // Create a session.
                        final Session consumerSession = consumerConnection
                                .createSession(false, Session.AUTO_ACKNOWLEDGE);

                        // Create a queue named "MyQueue".
                        final Destination consumerDestination = consumerSession
                                .createQueue("MyQueue");

                        // Create a message consumer from the session to the queue.
                        final MessageConsumer consumer = consumerSession
                                .createConsumer(consumerDestination);

                        // Begin to wait for messages.
                        final Message consumerMessage = consumer.receive(1000);

                        // Receive the message when it arrives.
                        final TextMessage consumerTextMessage = (TextMessage)
consumerMessage;
                        System.out.println("Message received: " +
consumerTextMessage.getText());

                        // Clean up the consumer.
                        consumer.close();
```

```
                        consumerSession.close();
                        consumerConnection.close();
                    }

                    private static PooledConnectionFactory
                    createPooledConnectionFactory(ActiveMQConnectionFactory
connectionFactory) {
                        // Create a pooled connection factory.
                        final PooledConnectionFactory pooledConnectionFactory =
                                new PooledConnectionFactory();

pooledConnectionFactory.setConnectionFactory(connectionFactory);
                        pooledConnectionFactory.setMaxConnections(10);
                        return pooledConnectionFactory;
                    }

                    private static ActiveMQConnectionFactory
createActiveMQConnectionFactory() {
                        // Create a connection factory.
                        final ActiveMQConnectionFactory connectionFactory =
                                new ActiveMQConnectionFactory(WIRE_LEVEL_ENDPOINT);

                        // Pass the sign-in credentials.
                        connectionFactory.setUserName(ACTIVE_MQ_USERNAME);
                        connectionFactory.setPassword(ACTIVE_MQ_PASSWORD);
                        return connectionFactory;
                    }
                    }
```

MQTT

```
/*
 * Copyright 2010-2019 Amazon.com, Inc. or its affiliates. All Rights Reserved.
 *
 * Licensed under the Apache License, Version 2.0 (the "License").
 * You may not use this file except in compliance with the License.
 * A copy of the License is located at
 *
 *  https://aws.amazon.com/apache2.0
 *
 * or in the "license" file accompanying this file. This file is distributed
 * on an "AS IS" BASIS, WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either
 * express or implied. See the License for the specific language governing
```

```
 * permissions and limitations under the License.
 *
 */

                    import org.eclipse.paho.client.mqttv3.*;

                    public class AmazonMQExampleMqtt implements MqttCallback {

                    // Specify the connection parameters.
                    private final static String WIRE_LEVEL_ENDPOINT =
                            "ssl://b-1234a5b6-78cd-901e-2fgh-3i45j6k178l9-1.mq.us-
east-2.amazonaws.com:8883";
                    private final static String ACTIVE_MQ_USERNAME =
 "MyUsername123";
                    private final static String ACTIVE_MQ_PASSWORD =
 "MyPassword456";

                    public static void main(String[] args) throws Exception {
                        new AmazonMQExampleMqtt().run();
                    }

                    private void run() throws MqttException, InterruptedException {

                        // Specify the topic name and the message text.
                        final String topic = "myTopic";
                        final String text = "Hello from Amazon MQ!";

                        // Create the MQTT client and specify the connection
 options.
                        final String clientId = "abc123";
                        final MqttClient client = new
 MqttClient(WIRE_LEVEL_ENDPOINT, clientId);
                        final MqttConnectOptions connOpts = new
 MqttConnectOptions();

                        // Pass the sign-in credentials.
                        connOpts.setUserName(ACTIVE_MQ_USERNAME);
                        connOpts.setPassword(ACTIVE_MQ_PASSWORD.toCharArray());

                        // Create a session and subscribe to a topic filter.
                        client.connect(connOpts);
                        client.setCallback(this);
                        client.subscribe("+");
```

```
                        // Create a message.
                        final MqttMessage message = new
 MqttMessage(text.getBytes());

                        // Publish the message to a topic.
                        client.publish(topic, message);
                        System.out.println("Published message.");

                        // Wait for the message to be received.
                        Thread.sleep(3000L);

                        // Clean up the connection.
                        client.disconnect();
                    }

                    @Override
                    public void connectionLost(Throwable cause) {
                        System.out.println("Lost connection.");
                    }

                    @Override
                    public void messageArrived(String topic, MqttMessage message)
 throws MqttException {
                        System.out.println("Received message from topic " + topic +
 ": " + message);
                    }

                    @Override
                    public void deliveryComplete(IMqttDeliveryToken token) {
                        System.out.println("Delivered message.");
                    }
                    }
```

STOMP+WSS

```
/*
 * Copyright 2010-2019 Amazon.com, Inc. or its affiliates. All Rights Reserved.
 *
 * Licensed under the Apache License, Version 2.0 (the "License").
 * You may not use this file except in compliance with the License.
 * A copy of the License is located at
 *
 *  https://aws.amazon.com/apache2.0
```

```
 *
 * or in the "license" file accompanying this file. This file is distributed
 * on an "AS IS" BASIS, WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either
 * express or implied. See the License for the specific language governing
 * permissions and limitations under the License.
 *
 */

                    import
 org.springframework.messaging.converter.StringMessageConverter;
                    import org.springframework.messaging.simp.stomp.*;
                    import org.springframework.web.socket.WebSocketHttpHeaders;
                    import org.springframework.web.socket.client.WebSocketClient;
                    import
 org.springframework.web.socket.client.standard.StandardWebSocketClient;
                    import
 org.springframework.web.socket.messaging.WebSocketStompClient;

                    import java.lang.reflect.Type;

                    public class AmazonMQExampleStompWss {

                    // Specify the connection parameters.
                    private final static String DESTINATION = "/queue";
                    private final static String WIRE_LEVEL_ENDPOINT =
                            "wss://b-1234a5b6-78cd-901e-2fgh-3i45j6k178l9-1.mq.us-
east-2.amazonaws.com:61619";
                    private final static String ACTIVE_MQ_USERNAME =
 "MyUsername123";
                    private final static String ACTIVE_MQ_PASSWORD =
 "MyPassword456";

                    public static void main(String[] args) throws Exception {
                        final AmazonMQExampleStompWss example = new
 AmazonMQExampleStompWss();

                        final StompSession stompSession = example.connect();
                        System.out.println("Subscribed to a destination using
 session.");
                        example.subscribeToDestination(stompSession);

                        System.out.println("Sent message to session.");
                        example.sendMessage(stompSession);
                        Thread.sleep(60000);
```

```
                    }

                    private StompSession connect() throws Exception {
                        // Create a client.
                        final WebSocketClient client = new
StandardWebSocketClient();
                        final WebSocketStompClient stompClient = new
WebSocketStompClient(client);
                        stompClient.setMessageConverter(new
StringMessageConverter());

                        final WebSocketHttpHeaders headers = new
WebSocketHttpHeaders();

                        // Create headers with authentication parameters.
                        final StompHeaders head = new StompHeaders();
                        head.add(StompHeaders.LOGIN, ACTIVE_MQ_USERNAME);
                        head.add(StompHeaders.PASSCODE, ACTIVE_MQ_PASSWORD);

                        final StompSessionHandler sessionHandler = new
MySessionHandler();

                        // Create a connection.
                        return stompClient.connect(WIRE_LEVEL_ENDPOINT, headers,
head,
                                sessionHandler).get();
                    }

                    private void subscribeToDestination(final StompSession
stompSession) {
                        stompSession.subscribe(DESTINATION, new MyFrameHandler());
                    }

                    private void sendMessage(final StompSession stompSession) {
                        stompSession.send(DESTINATION, "Hello from Amazon
MQ!".getBytes());
                    }

                    private static class MySessionHandler extends
StompSessionHandlerAdapter {
                        public void afterConnected(final StompSession stompSession,
                                                   final StompHeaders stompHeaders) {
                            System.out.println("Connected to broker.");
                        }
```

```
                    }

                    private static class MyFrameHandler implements StompFrameHandler
{
                        public Type getPayloadType(final StompHeaders headers) {
                            return String.class;
                        }

                        public void handleFrame(final StompHeaders stompHeaders,
                                                final Object message) {
                            System.out.print("Received message from topic: " +
message);
                        }
                    }
                    }
```

Mengelola versi mesin Amazon MQ for ActiveMQ

Apache ActiveMQ mengatur nomor versi sesuai dengan spesifikasi versioning semantik sebagai `X.Y.Z`. Di Amazon MQ untuk implementasi ActiveMQ, `X` menunjukkan versi utama, `Y` mewakili versi minor, dan menunjukkan nomor versi patch. `Z` Amazon MQ menganggap perubahan versi sebagai utama jika nomor versi utama berubah. Misalnya, memutakhirkan dari versi 5 .17 ke 6.0 dianggap sebagai peningkatan versi utama. Perubahan versi dianggap kecil jika hanya nomor versi minor atau patch yang berubah. Misalnya, memutakhirkan dari versi 5. 18 hingga 5. 19 dianggap sebagai upgrade versi minor. Saat `autoMinorVersionUpgrade` dihidupkan, Amazon MQ meningkatkan broker Anda ke versi patch terbaru yang tersedia.

Amazon MQ untuk ActiveMQ merekomendasikan semua broker menggunakan versi minor terbaru yang didukung. Untuk petunjuk tentang cara memutakhirkan versi mesin broker Anda, lihat Memutakhirkan versi mesin broker Amazon MQ.

Versi mesin yang didukung di Amazon MQ untuk ActiveMQ

Kalender dukungan versi Amazon MQ menunjukkan kapan versi mesin broker akan mencapai akhir dukungan. Ketika versi mencapai akhir dukungan, Amazon MQ meningkatkan semua broker pada versi ini ke versi yang didukung berikutnya secara otomatis. Upgrade ini berlangsung selama jendela pemeliharaan terjadwal broker Anda, dalam 45 hari setelah end-of-support tanggal.

Amazon MQ menyediakan setidaknya pemberitahuan 90 hari sebelum versi mencapai akhir dukungan. Kami merekomendasikan untuk meningkatkan broker Anda sebelum end-of-support

tanggal untuk mencegah gangguan apa pun. Selain itu, Anda tidak dapat membuat broker baru pada versi yang dijadwalkan untuk akhir dukungan dalam waktu 30 hari sejak akhir tanggal dukungan.

Versi ActiveMQ Apache	Akhir dukungan di Amazon MQ
ActiveMQ 5.19 (disarankan)	
ActiveMQ 5.18	
ActiveMQ 5.17	Juni 16, 2025
ActiveMQ 5.16	November 15, 2024
ActiveMQ 5.15	September 16, 2024

Ketika membuat broker baru Amazon MQ for ActiveMQ, Anda dapat menentukan versi mesin ActiveMQ yang didukung. Jika Anda tidak menentukan nomor versi mesin saat membuat broker, Amazon MQ secara otomatis default ke nomor versi mesin terbaru.

Peningkatan versi mesin

Anda dapat meningkatkan broker Anda secara manual kapan saja ke versi mayor atau minor yang didukung berikutnya. Saat Anda mengaktifkan upgrade versi minor otomatis, Amazon MQ akan meningkatkan broker Anda ke versi patch terbaru yang didukung selama jendela pemeliharaan.

Untuk informasi lebih lanjut tentang meningkatkan broker Anda secara manual, lihatthe section called “Meningkatkan versi mesin”.

Membuat daftar versi mesin yang didukung

Anda dapat membuat daftar semua versi mesin minor dan utama yang didukung dengan menggunakan `describe-broker-instance-options` AWS CLI perintah.

```
aws mq describe-broker-instance-options
```

Untuk memfilter hasil menurut mesin dan jenis instans, gunakan opsi `--engine-type` dan `--host-instance-type` seperti yang ditampilkan di bawah.

```
aws mq describe-broker-instance-options --engine-type engine-type --host-instance-
type instance-type
```

Misalnya, untuk memfilter hasil ActiveMQ, dan jenis instance, *engine-type* ganti `ACTIVEMQ` dengan `mq.m5.large` dan dengan. *instance-type* `mq.m5.large`

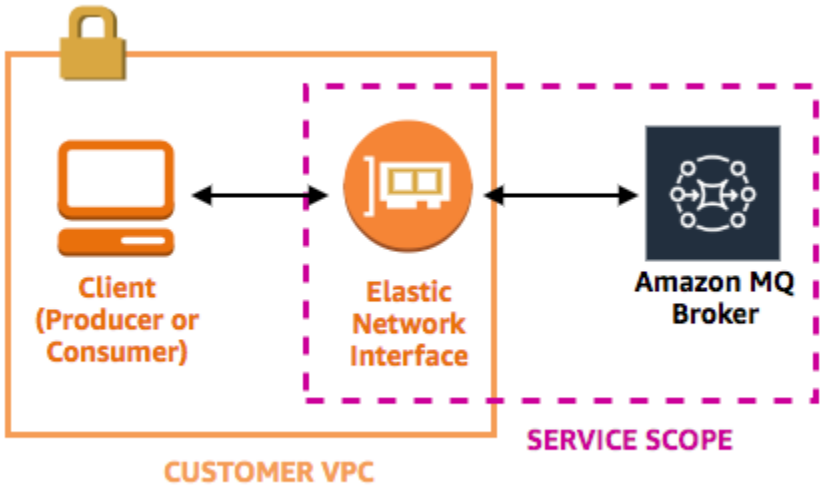
Amazon MQ untuk praktik terbaik ActiveMQ

Gunakan ini sebagai referensi untuk menemukan rekomendasi dengan cepat guna memaksimalkan performa dan meminimalkan biaya throughput saat bekerja dengan broker ActiveMQ di Amazon MQ.

Jangan Pernah Memodifikasi atau Menghapus Antarmuka Jaringan Elastis Amazon MQ

Ketika Anda pertama kali membuat broker Amazon MQ, Amazon MQ menyediakan antarmuka jaringan elastis pada Virtual Private Cloud (VPC) di bawah akun Anda dan memerlukan sejumlah izin EC2. Antarmuka jaringan memungkinkan klien Anda (produsen atau konsumen) berkomunikasi dengan broker Amazon MQ. Antarmuka jaringan dianggap berada dalam lingkup layanan Amazon MQ, meski merupakan bagian dari VPC akun Anda.

> **Warning**
>
> Anda tidak harus memodifikasi atau menghapus antarmuka jaringan ini. Memodifikasi atau menghapus antarmuka jaringan dapat menyebabkan koneksi hilang permanen antara VPC dan broker Anda.



Selalu Gunakan Pooling Koneksi

Dalam skenario dengan produsen tunggal dan konsumen tunggal (seperti tutorial Memulai: Membuat dan menghubungkan ke broker ActiveMQ), Anda dapat menggunakan satu kelas `ActiveMQConnectionFactory` untuk setiap produsen dan konsumen. Sebagai contoh:

```
// Create a connection factory.
final ActiveMQConnectionFactory connectionFactory = new
 ActiveMQConnectionFactory(wireLevelEndpoint);

// Pass the sign-in credentials.
connectionFactory.setUserName(activeMqUsername);
connectionFactory.setPassword(activeMqPassword);

// Establish a connection for the consumer.
final Connection consumerConnection = connectionFactory.createConnection();
consumerConnection.start();
```

Namun, dalam skenario yang lebih realistis dengan beberapa produsen dan konsumen, membuat sejumlah besar koneksi untuk beberapa produsen dapat menghabiskan banyak biaya. Dalam skenario ini, Anda harus mengelompokkan beberapa permintaan produsen menggunakan kelas `PooledConnectionFactory`. Sebagai contoh:

> **Note**
>
> Konsumen pesan jangan pernah gunakan kelas `PooledConnectionFactory`.

```
// Create a connection factory.
final ActiveMQConnectionFactory connectionFactory = new
 ActiveMQConnectionFactory(wireLevelEndpoint);

// Pass the sign-in credentials.
connectionFactory.setUserName(activeMqUsername);
connectionFactory.setPassword(activeMqPassword);

// Create a pooled connection factory.
final PooledConnectionFactory pooledConnectionFactory = new PooledConnectionFactory();
pooledConnectionFactory.setConnectionFactory(connectionFactory);
pooledConnectionFactory.setMaxConnections(10);

// Establish a connection for the producer.
final Connection producerConnection = pooledConnectionFactory.createConnection();
producerConnection.start();
```

Selalu Gunakan Transportasi Failover untuk Terhubung ke Beberapa Titik Akhir Broker

Jika aplikasi Anda perlu terhubung ke beberapa titik akhir broker—misalnya, ketika Anda menggunakan mode deployment aktif/siaga atau saat Anda bermigrasi dari broker pesan on-premise ke Amazon MQ—gunakan Transportasi Failover untuk mengizinkan konsumen Anda terhubung secara acak ke salah satu titik akhir. Contoh:

```
failover:(ssl://b-1234a5b6-78cd-901e-2fgh-3i45j6k178l9-1.mq.us-
east-2.amazonaws.com:61617,ssl://b-9876l5k4-32ji-109h-8gfe-7d65c4b132a1-2.mq.us-
west-2.amazonaws.com:61617)?randomize=true
```

> **Important**
>
> Broker zona multi-ketersediaan dapat mengalami kegagalan selama jendela pemeliharaan dan restart broker. Gunakan Failover Transport untuk memastikan ketersediaan broker Anda.

Hindari Penggunaan Penyeleksi Pesan

Anda dapat menggunakan penyeleksi JMS untuk melampirkan filter ke langganan topik (untuk merutekan pesan ke konsumen berdasarkan kontennya). Namun, penggunaan penyeleksi JMS memenuhi buffer filter broker Amazon MQ, mencegahnya memfilter pesan.

Secara umum, buat agar konsumen tidak dapat merutekan pesan karena, untuk pemisahan yang optimal antara konsumen dan produsen, baik konsumen dan produsen harus bersifat sementara.

Memilih Tujuan Virtual untuk Langganan Tahan Lama

Langganan tahan lama dapat membantu memastikan bahwa konsumen menerima semua pesan yang dipublikasikan ke topik, misalnya, setelah koneksi yang hilang dipulihkan. Namun, penggunaan langganan tahan lama juga menghalangi penggunaan konsumen yang bersaing dan mungkin memiliki masalah performa dalam skala besar. Pertimbangkan untuk menggunakan tujuan virtual.

Jika menggunakan peering VPC Amazon, hindari klien IPs dalam rentang CIDR **`10.0.0.0/16`**

Jika Anda menyiapkan peering VPC Amazon antara infrastruktur on-premise dan broker Amazon MQ Anda, Anda tidak boleh mengonfigurasi koneksi klien dengan rentang CIDR. IPs `10.0.0.0/16`

Menonaktifkan Penyimpanan dan Pengiriman Bersamaan untuk Antrean dengan Konsumen Lambat

Secara default, Amazon MQ mengoptimalkan antrean dengan konsumen cepat:

- Konsumen dianggap cepat jika mereka mampu bersaing dengan laju pesan yang dihasilkan oleh produsen.
- Konsumen dianggap lambat jika antrean menimbulkan backlog pesan yang tidak diakui, berpotensi menyebabkan penurunan throughput produsen.

Untuk menginstruksikan Amazon MQ agar mengoptimalkan antrean dengan konsumen lambat, atur `concurrentStoreAndDispatchQueues` atribut ke `false`. Contoh konfigurasi, lihat `concurrentStoreAndDispatchQueues`.

Memilih Tipe Instans Broker yang Tepat untuk Throughput Terbaik

Throughput pesan dari tipe instans broker tergantung pada kasus penggunaan aplikasi Anda dan faktor berikut:

- Penggunaan ActiveMQ dalam mode tetap
- Ukuran pesan
- Jumlah produsen dan konsumen
- Jumlah tujuan

Memahami hubungan antara ukuran pesan, latensi, dan throughput

Tergantung pada kasus penggunaan Anda, tipe instans broker yang lebih besar mungkin tidak selalu meningkatkan throughput sistem. Ketika ActiveMQ menulis pesan ke penyimpanan tahan lama, ukuran pesan Anda menentukan faktor pembatas sistem:

- Jika pesan Anda lebih kecil dari 100 KB, latensi penyimpanan tetap adalah faktor pembatas.
- Jika pesan Anda lebih besar dari 100 KB, throughput penyimpanan tetap adalah faktor pembatas.

Ketika Anda menggunakan ActiveMQ dalam mode tetap, menulis ke penyimpanan biasanya terjadi ketika ada beberapa konsumen atau ketika konsumen lambat. Dalam modus tidak tetap, menulis ke penyimpanan juga terjadi dengan konsumen lambat jika memori tumpukan instans broker penuh.

Untuk menentukan tipe instans broker terbaik bagi aplikasi Anda, kami merekomendasikan pengujian tipe instans broker yang berbeda. Untuk informasi selengkapnya, lihat Broker instance types dan juga Mengukur Throughput untuk Amazon MQ menggunakan Tolok Ukur JMS.

Kasus penggunaan untuk jenis instans broker yang lebih besar

Ada tiga kasus penggunaan umum ketika tipe instans broker yang lebih besar meningkatkan throughput:

- Mode tidak tetap – Ketika aplikasi Anda kurang sensitif terhadap kehilangan pesan selama failover instans broker (misalnya, ketika menyiarkan skor olahraga), Anda mungkin sering menggunakan mode tidak tetap ActiveMQ. Dalam mode ini, ActiveMQ menulis pesan ke penyimpanan tetap hanya jika memori tumpukan instans broker penuh. Sistem yang menggunakan mode tidak tetap bisa mendapatkan manfaat dari jumlah memori yang lebih tinggi, CPU yang lebih cepat, dan jaringan yang lebih cepat dan tersedia pada tipe instans broker yang lebih besar.

- Konsumen cepat – Ketika konsumen aktif tersedia dan bendera `concurrentStoreAndDispatchQueues` diaktifkan, ActiveMQ memungkinkan pesan mengalir langsung dari produsen ke konsumen tanpa mengirim pesan ke penyimpanan (bahkan dalam mode tetap). Jika aplikasi Anda dapat mengonsumsi pesan dengan cepat (atau jika Anda dapat merancang konsumen Anda untuk melakukan hal ini), aplikasi bisa mendapatkan keuntungan dari tipe instans broker yang lebih besar. Untuk membiarkan aplikasi Anda mengonsumsi pesan lebih cepat, tambahkan utas konsumen ke instans aplikasi Anda atau tingkatkan skala instans aplikasi Anda secara vertikal atau horizontal.
- Transaksi yang di-batch – Ketika menggunakan mode tetap dan mengirim beberapa pesan per transaksi, Anda dapat mencapai throughput pesan yang lebih tinggi secara keseluruhan dengan menggunakan tipe instans broker yang lebih besar. Untuk informasi selengkapnya, lihat Should I Use Transactions? dalam dokumentasi ActiveMQ.

Pilih jenis penyimpanan broker yang tepat untuk throughput terbaik

Untuk memanfaatkan daya tahan dan replikasi yang tinggi di beberapa Availability Zone, gunakan Amazon EFS. Untuk memanfaatkan latensi rendah dan throughput yang tinggi, gunakan Amazon EBS. Untuk informasi selengkapnya, lihat Storage.

Mengonfigurasi Jaringan Broker dengan Benar

Saat Anda membuat jaringan broker, konfigurasikan dengan benar untuk aplikasi Anda:

- Aktifkan mode tetap – Karena (tergantung pada rekannya) setiap instans broker bertindak seperti produsen atau konsumen, jaringan broker tidak menyediakan replikasi terdistribusi dari pesan. Broker pertama yang bertindak sebagai konsumen menerima pesan dan menahannya ke penyimpanan. Broker ini mengirimkan pengakuan kepada produsen dan meneruskan pesan ke broker berikutnya. Ketika broker kedua mengakui ketetapan pesan, broker pertama akan menghapus pesan.

  Jika modus tetap dinonaktifkan, broker pertama mengakui produsen tanpa menahan pesan ke penyimpanan. Untuk informasi selengkapnya, lihat Replicated Message Store dan What is the difference between persistent and non-persistent delivery? dalam dokumentasi Apache ActiveMQ.
- Jangan nonaktifkan pesan penasihat untuk instans broker – Untuk informasi selengkapnya, lihat Advisory Message dalam dokumentasi Apache ActiveMQ.

- Jangan gunakan penemuan broker multicast – Amazon MQ tidak mendukung penemuan broker menggunakan multicast. Untuk informasi selengkapnya, lihat What is the difference between discovery, multicast, and zeroconf? dalam dokumentasi Apache ActiveMQ.

Hindari mulai ulang lambat dengan memulihkan transaksi XA yang disiapkan

ActiveMQ mendukung transaksi terdistribusi (XA). Mengetahui cara ActiveMQ memproses transaksi XA dapat membantu menghindari waktu pemulihan yang lambat untuk mulai ulang dan failover broker di Amazon MQ

Transaksi XA yang disiapkan dan belum terselesaikan akan diputar ulang pada setiap mulai ulang. Jika masih belum terselesaikan, jumlahnya akan bertambah dari waktu ke waktu, secara signifikan meningkatkan waktu yang dibutuhkan untuk memulai broker. Hal ini memengaruhi waktu mulai ulang dan failover. Anda harus menyelesaikan transaksi ini dengan `commit()` atau `rollback()` agar performa tidak menurun seiring waktu.

Untuk memantau transaksi XA yang belum terselesaikan, Anda dapat menggunakan `JournalFilesForFastRecovery` metrik di Amazon CloudWatch Logs. Jika jumlah ini meningkat, atau secara konsisten lebih tinggi dari 1, Anda harus memulihkan transaksi yang belum terselesaikan dengan kode yang serupa dengan contoh berikut. Untuk informasi selengkapnya, lihat Kuota di Amazon MQ.

Kode contoh berikut berjalan menelusuri transaksi XA yang disiapkan dan menutupnya dengan `rollback()`.

```
import org.apache.activemq.ActiveMQXAConnectionFactory;

import javax.jms.XAConnection;
import javax.jms.XASession;
import javax.transaction.xa.XAResource;
import javax.transaction.xa.Xid;

public class RecoverXaTransactions {
    private static final ActiveMQXAConnectionFactory ACTIVE_MQ_CONNECTION_FACTORY;
    final static String WIRE_LEVEL_ENDPOINT =
            "tcp://localhost:61616";;
    static {
        final String activeMqUsername = "MyUsername123";
        final String activeMqPassword = "MyPassword456";
```

```
        ACTIVE_MQ_CONNECTION_FACTORY = new
 ActiveMQXAConnectionFactory(activeMqUsername, activeMqPassword, WIRE_LEVEL_ENDPOINT);
        ACTIVE_MQ_CONNECTION_FACTORY.setUserName(activeMqUsername);
        ACTIVE_MQ_CONNECTION_FACTORY.setPassword(activeMqPassword);
    }

    public static void main(String[] args) {
        try {
            final XAConnection connection =
 ACTIVE_MQ_CONNECTION_FACTORY.createXAConnection();
            XASession xaSession = connection.createXASession();
            XAResource xaRes = xaSession.getXAResource();

            for (Xid id : xaRes.recover(XAResource.TMENDRSCAN)) {
                xaRes.rollback(id);
            }
            connection.close();

        } catch (Exception e) {
        }
    }
}
```

Dalam skenario dunia nyata, Anda dapat memeriksa transaksi XA yang disiapkan pada Manajer Transaksi XA. Kemudian Anda dapat memutuskan apakah akan menangani setiap transaksi yang disiapkan dengan `rollback()` atau `commit()`.

Menggunakan Amazon MQ untuk RabbitMQ

Amazon MQ memudahkan pembuatan broker pesan dengan sumber daya komputasi dan penyimpanan yang sesuai dengan kebutuhan Anda. Anda dapat membuat, mengelola, dan menghapus broker menggunakan Konsol Manajemen AWS, Amazon MQ REST API, atau. AWS Command Line Interface

Bagian ini menjelaskan elemen dasar broker pesan untuk jenis mesin ActiveMQ dan RabbitMQ, daftar tipe instans broker Amazon MQ yang tersedia dan statusnya, serta memberikan gambaran umum tentang arsitektur broker juga opsi konfigurasi.

Untuk mempelajari tentang Amazon MQ REST APIs, lihat Referensi API Amazon MQ REST.

Apa itu Amazon MQ untuk broker RabbitMQ?

Broker adalah lingkungan broker pesan yang berjalan di Amazon MQ. Ini adalah blok bangunan dasar Amazon MQ. Deskripsi gabungan dari kelas instance broker (`m7g`) dan size (`large`,`medium`) disebut tipe instance broker (misalnya,`mq.m7g.large`).

- Broker single instance terdiri dari satu broker dalam satu Availability Zone di belakang Network Load Balancer (NLB). Broker berkomunikasi dengan aplikasi Anda dan dengan volume penyimpanan Amazon EBS.
- Deployment klaster adalah pengelompokan logis dari tiga node broker RabbitMQ di balik Penyeimbang Beban Jaringan, masing-masing membagikan pengguna, antrean, dan status terdistribusi di beberapa Availability Zone (AZ).

Untuk informasi lebih lanjut, lihat Menerapkan broker RabbitMQ.

Port listener

Broker RabbitMQ yang dikelola Amazon MQ mendukung port pendengar berikut untuk konektivitas tingkat aplikasi melalui. `amqps` Anda juga dapat menggunakan port ini untuk koneksi klien menggunakan konsol web RabbitMQ dan API manajemen. Semua koneksi menggunakan enkripsi TLS untuk keamanan.

- Port pendengar `5671` - Digunakan untuk koneksi AMQP aman yang dibuat melalui URL AMQP yang aman. Port ini mendukung protokol AMQP 0-9-1 dan AMQP 1.0 di RabbitMQ 4. Sebagai

contoh, broker dengan ID broker `b-c8352341-ec91-4a78-ad9c-a43f23d325bb`, di-deploy di wilayah `us-west-2`, berikut adalah URL lengkap `amqps` broker: `b-c8352341-ec91-4a78-ad9c-a43f23d325bb.mq.us-west-2.amazonaws.com:5671`.

- Port pendengar `443` dan `15671` - Anda dapat menggunakan kedua port pendengar secara bergantian untuk mengakses broker melalui konsol web RabbitMQ atau API manajemen. Port 443 menyediakan akses HTTPS standar, sedangkan port 15671 adalah port manajemen RabbitMQ tradisional dengan enkripsi TLS.

Atribut

Broker RabbitMQ memiliki beberapa atribut:

- Nama. Misalnya, `MyBroker`.
- ID. Misalnya, `b-1234a5b6-78cd-901e-2fgh-3i45j6k178l9`.
- Amazon Resource Name (ARN). Misalnya, `arn:aws:mq:us-east-2:123456789012:broker:MyBroker:b-1234a5b6-78cd-901e-2fgh-3i45j6k178l9`.
- URL konsol web RabbitMQ. Misalnya, `https://b-1234a5b6-78cd-901e-2fgh-3i45j6k178l9-1.mq.us-east-2.amazonaws.com`.

  Untuk informasi selengkapnya, lihat konsol web RabbitMQ dalam dokumentasi RabbitMQ.
- Titik akhir aman AMQP. Misalnya, `amqps://b-1234a5b6-78cd-901e-2fgh-3i45j6k178l9-1.mq.us-east-2.amazonaws.com`.

Untuk daftar lengkap atribut broker, lihat di Referensi REST API Amazon MQ:

- ID Operasi REST: Broker
- ID Operasi REST: Broker
- ID Operasi REST: Reboot Broker

Mengelola Amazon MQ untuk versi mesin RabbitMQ

RabbitMQ mengatur nomor versi sesuai dengan spesifikasi versi semantik sebagai. `X.Y.Z` Di Amazon MQ untuk implementasi RabbitMQ, `X` menunjukkan versi utama, `Y` mewakili versi minor, dan menunjukkan nomor versi patch. `Z` Amazon MQ menganggap perubahan versi sebagai utama jika nomor versi utama berubah. Misalnya, memutakhirkan dari versi 3 .13 ke 4 .0 dianggap sebagai

peningkatan versi utama. Perubahan versi dianggap kecil jika hanya nomor versi minor atau patch yang berubah. Misalnya, memutakhirkan dari versi 3. 11 .28 hingga 3. 12.13 dianggap sebagai peningkatan versi minor.

Amazon MQ untuk RabbitMQ merekomendasikan semua broker menggunakan versi terbaru yang didukung RabbitMQ 4.2. Untuk petunjuk tentang cara meningkatkan versi mesin broker Anda, lihat Memutakhirkan versi mesin broker Amazon MQ.

Saat Anda membuat Amazon MQ baru untuk broker RabbitMQ, Anda hanya perlu menentukan nomor versi mayor dan minor. Misalnya, RabbitMQ 4.2. Jika Anda tidak menentukan versi mesin saat membuat broker, Amazon MQ secara otomatis default ke versi mesin terbaru.

> **Important**
>
> Amazon MQ tidak mendukung streaming. Membuat aliran akan mengakibatkan hilangnya data.
> Amazon MQ tidak mendukung penggunaan logging terstruktur di JSON.

Amazon MQ mendukung dua rilis versi utama RabbitMQ:

- KelinciMQ 4

  Amazon MQ mendukung RabbitMQ 4.2 dalam seri rilis RabbitMQ 4 hanya pada jenis instans mq.m7g di semua ukuran instans yang didukung.
- KelinciMQ 3

  Amazon MQ mendukung RabbitMQ 3.13 dalam seri rilis RabbitMQ 3 pada jenis instans mq.t3, mq.m5, dan mq.m7g di semua ukuran instans yang didukung.

Membuat daftar versi mesin yang didukung

Anda dapat membuat daftar semua versi mesin minor dan utama yang didukung dengan menggunakan `describe-broker-instance-options` AWS CLI perintah.

```
aws mq describe-broker-instance-options
```

Untuk memfilter hasil menurut mesin dan jenis instans, gunakan opsi `--engine-type` dan `--host-instance-type` seperti yang ditampilkan di bawah.

```
aws mq describe-broker-instance-options --engine-type engine-type --host-instance-
type instance-type
```

Misalnya, untuk memfilter hasil untuk RabbitMQ, dan jenis `mq.m7g.large` instance, ganti *`engine-type`* dengan dan dengan`RABBITMQ`. *`instance-type`* `mq.m7g.large`

KelinciMQ 4

Amazon MQ mendukung RabbitMQ 4.2 dalam seri rilis RabbitMQ 4 hanya pada jenis instans mq.m7g di semua ukuran instans yang didukung.

> **Important**
>
> Anda hanya dapat membuat broker baru di RabbitMQ 4.2. Upgrade di tempat dari RabbitMQ 3.13 saat ini tidak didukung.

> **Important**
>
> Jenis antrian default di Amazon MQ untuk broker RabbitMQ 4.2 adalah “kuorum”. Jika tidak ada argumen tipe antrian yang ditentukan selama pembuatan antrian, antrian kuorum akan dibuat.
> Kami sangat merekomendasikan penggunaan antrian kuorum pada RabbitMQ 4 untuk kebutuhan daya tahan, karena antrian klasik tidak dijamin tahan lama dalam semua kasus.

Perubahan berikut telah diperkenalkan di RabbitMQ 4 di Amazon MQ

- AMQP 1.0 sebagai protokol inti: Untuk informasi selengkapnya, lihat Protokol.
- Sekop lokal: Sekop sekarang mendukung protokol baru yang disebut “lokal” selain AMQP 0-9-1 dan AMQP 1.0. Sekop lokal secara internal didasarkan pada AMQP 1.0 tetapi alih-alih menggunakan koneksi TCP terpisah, mereka menggunakan koneksi intra-cluster antara node cluster dan internal untuk menerbitkan dan mengkonsumsi pesan. APIs Ini hanya dapat digunakan untuk mengkonsumsi dan menerbitkan dalam cluster yang sama dan dapat menawarkan throughput yang lebih tinggi saat menggunakan sumber daya yang lebih sedikit daripada AMQP 0-9-1 dan AMQP 1.0.

- Antrian kuorum mendukung prioritas pesan: Prioritas pesan antrian kuorum selalu aktif dan tidak memerlukan kebijakan untuk bekerja. Segera setelah antrian kuorum menerima pesan dengan set prioritas, itu akan memungkinkan prioritas. Antrian kuorum secara internal hanya mendukung dua prioritas - tinggi dan normal. Pesan tanpa set prioritas akan dipetakan ke normal seperti halnya prioritas 0 - 4. Pesan dengan prioritas lebih tinggi dari 4 akan dipetakan ke tinggi. Pesan prioritas tinggi akan lebih disukai daripada pesan prioritas normal dengan rasio 2:1, yaitu untuk setiap 2 pesan prioritas tinggi, antrian akan mengirimkan 1 pesan prioritas normal (jika tersedia). Oleh karena itu, antrian kuorum menerapkan semacam pemrosesan prioritas “pembagian yang adil” yang tidak ketat. Ini memastikan kemajuan selalu dibuat pada pesan prioritas normal, tetapi prioritas tinggi disukai pada rasio 2:1.
- Khepri: Khepri digunakan sebagai toko metadata default untuk broker RabbitMQ 4
- Mutual TLS (MTLS): Amazon MQ mendukung TLS bersama (MTLS) untuk broker RabbitMQ, memungkinkan klien untuk mengautentikasi menggunakan sertifikat. Untuk informasi selengkapnya, lihat konfigurasi mTLS.
- Plugin otentikasi sertifikat SSL: Plugin otentikasi SSL menggunakan sertifikat klien dari koneksi mTLS untuk mengautentikasi pengguna, memungkinkan otentikasi menggunakan sertifikat klien X.509 alih-alih kredensyal nama pengguna dan kata sandi. Untuk informasi selengkapnya, lihat otentikasi sertifikat SSL.
- Plugin otentikasi HTTP: Plugin backend otentikasi HTTP memungkinkan pendelegasian otentikasi dan otorisasi ke layanan HTTP eksternal. Untuk informasi selengkapnya, lihat otentikasi dan otorisasi HTTP.
- Dukungan JMS: Broker sekarang mendukung beban kerja JMS dengan plugin pertukaran topik JMS diaktifkan, memungkinkan aplikasi JMS untuk terhubung menggunakan klien RabbitMQ JMS.

Fitur-fitur berikut tidak digunakan lagi dari RabbitMQ 4 di Amazon MQ

- Mencerminkan antrian klasik: Antrian klasik terus didukung tanpa ada perubahan yang merusak untuk pustaka dan aplikasi klien, tetapi sekarang merupakan tipe antrian yang tidak direplikasi. Klien akan dapat terhubung ke node mana pun untuk dipublikasikan dan dikonsumsi dari antrian klasik yang tidak direplikasi. Antrian kuorum direkomendasikan untuk replikasi dan keamanan data.
- Penghapusan QoS Global: Pelanggan disarankan untuk mengatur QoS per konsumen (non-global) alih-alih QoS Global, di mana satu prefetch bersama digunakan untuk seluruh saluran.
- Support untuk antrian transien dan non-eksklusif: Antrian sementara adalah antrian yang masa pakainya terkait dengan waktu aktif node tempat mereka dideklarasikan. Dalam satu contoh broker, mereka dihapus ketika node dimulai ulang. Dalam penerapan cluster, mereka dihapus ketika node

tempat mereka di-host dimulai ulang. Sebaiknya gunakan antrian TTL untuk menghapus antrian idle yang tidak digunakan secara otomatis setelah beberapa saat tidak aktif. Antrian eksklusif terus didukung dan dihapus setelah semua koneksi ke antrian telah dihapus.

Perubahan yang melanggar berikut dapat memengaruhi aplikasi Anda saat meningkatkan ke RabbitMQ 4.2 di Amazon MQ

- Jenis antrian default: Jenis antrian default pada broker RabbitMQ 4 diatur ke kuorum. Jika tidak ada argumen tipe antrian yang ditentukan selama pembuatan antrian, antrian kuorum akan dibuat.
- Batas pengiriman ulang default pada antrian kuorum disetel ke 20: Pesan yang dikirim ulang 20 kali atau lebih akan diberi huruf mati atau dibatalkan (dihapus). Jika 20 pengiriman per pesan adalah skenario umum untuk antrian, target huruf mati atau batas yang lebih tinggi harus dikonfigurasi untuk antrian tersebut untuk menghindari kehilangan data. Cara yang disarankan untuk melakukannya adalah melalui kebijakan.
- amqplib: Versi amqplib klien Node JS yang lebih lama dari 0.10.7 atau pustaka klien AMQP apa pun yang menggunakan frame_max < 8192 tidak akan dapat terhubung ke RabbitMQ
- Batas sumber daya default: Amazon MQ untuk RabbitMQ telah memperkenalkan batas penggunaan sumber daya default untuk koneksi, saluran, konsumen per saluran, antrian, vhost, sekop, pertukaran, dan ukuran pesan maksimum. Ini berfungsi sebagai pagar pembatas untuk melindungi ketersediaan broker dan dapat disesuaikan menggunakan konfigurasi agar sesuai dengan kebutuhan spesifik Anda.

Fitur-fitur berikut tidak didukung pada RabbitMQ 4 di Amazon MQ

- Pertukaran acak lokal: Pertukaran acak lokal tidak didukung di Amazon MQ karena node Amazon MQ berada di belakang penyeimbang beban jaringan.
- Pencegat Pesan: Pencegat pesan RabbitMQ tidak didukung di Amazon MQ.
- Metrik per antrian: Amazon MQ tidak akan menjual metrik antrian RabbitMQ untuk broker RabbitMQ 4 melalui. AWS CloudWatch Amazon MQ masih akan memberikan metrik tingkat broker melalui. AWS CloudWatch Anda dapat melakukan kueri metrik antrian menggunakan API manajemen RabbitMQ. Sebaiknya kueri metrik untuk antrian tertentu pada frekuensi interval satu menit atau lebih lama.

Dukungan versi RabbitMQ

Kalender dukungan versi Amazon MQ di bawah ini menunjukkan kapan versi mesin broker akan mencapai akhir dukungan. Ketika versi mencapai akhir dukungan, Amazon MQ meningkatkan semua broker pada versi ini ke versi yang didukung berikutnya secara otomatis. Upgrade ini berlangsung selama jendela pemeliharaan terjadwal broker Anda, dalam waktu 45 hari setelah end-of-support tanggal.

Amazon MQ menyediakan setidaknya pemberitahuan 90 hari sebelum versi mencapai akhir dukungan. Kami merekomendasikan untuk meningkatkan broker Anda sebelum end-of-support tanggal untuk mencegah gangguan apa pun. Selain itu, Anda tidak dapat membuat broker baru pada versi yang dijadwalkan untuk akhir dukungan dalam waktu 30 hari sejak akhir tanggal dukungan.

Versi RabbitMQ	Akhir dukungan di Amazon MQ
4.2 (Direkomendasikan)	
3.13	
3.12	Maret 17, 2025

Upgrade versi

Anda dapat meningkatkan broker Anda secara manual kapan saja ke versi mayor atau minor yang didukung berikutnya. Untuk informasi selengkapnya tentang meningkatkan broker Anda secara manual, lihat Memutakhirkan versi mesin broker Amazon MQ.

Amazon MQ mengelola peningkatan ke versi patch terbaru yang didukung untuk semua broker RabbitMQ menggunakan versi 3.13 ke atas. Peningkatan versi manual dan otomatis terjadi selama jendela pemeliharaan terjadwal atau setelah Anda melakukan boot ulang broker.

> **Important**
>
> RabbitMQ hanya mengizinkan pembaruan versi tambahan (mis: 3.9.x hingga 3.10.x). Anda tidak dapat melewati versi minor saat memperbarui (mis: 3.8.x ke 3.11.x).

Pialang instans tunggal akan offline saat di-boot ulang. Untuk broker cluster, antrian cermin harus disinkronkan selama reboot. Dengan antrian yang lebih panjang, proses sinkronisasi antrian bisa memakan waktu lebih lama. Selama proses sinkronisasi antrian, antrian tidak tersedia untuk konsumen dan produsen. Ketika proses sinkronisasi antrian selesai, broker menjadi tersedia lagi. Untuk meminimalkan dampak, kami sarankan untuk meningkatkan selama waktu lalu lintas rendah. Untuk informasi selengkapnya tentang praktik terbaik untuk peningkatan versi, lihatAmazon MQ untuk praktik terbaik RabbitMQ.

Opsi penyebaran untuk Amazon MQ untuk broker RabbitMQ

Broker RabbitMQ dapat dibuat sebagai broker instans tunggal atau dalam deployment klaster. Untuk kedua mode deployment, Amazon MQ memberikan daya tahan tinggi dengan menyimpan data secara redundan.

Anda dapat mengakses broker RabbitMQ menggunakan bahasa pemrograman yang didukung RabbitMQ dan dengan mengaktifkan TLS untuk protokol berikut:

- AMQP (0-9-1)

Topik

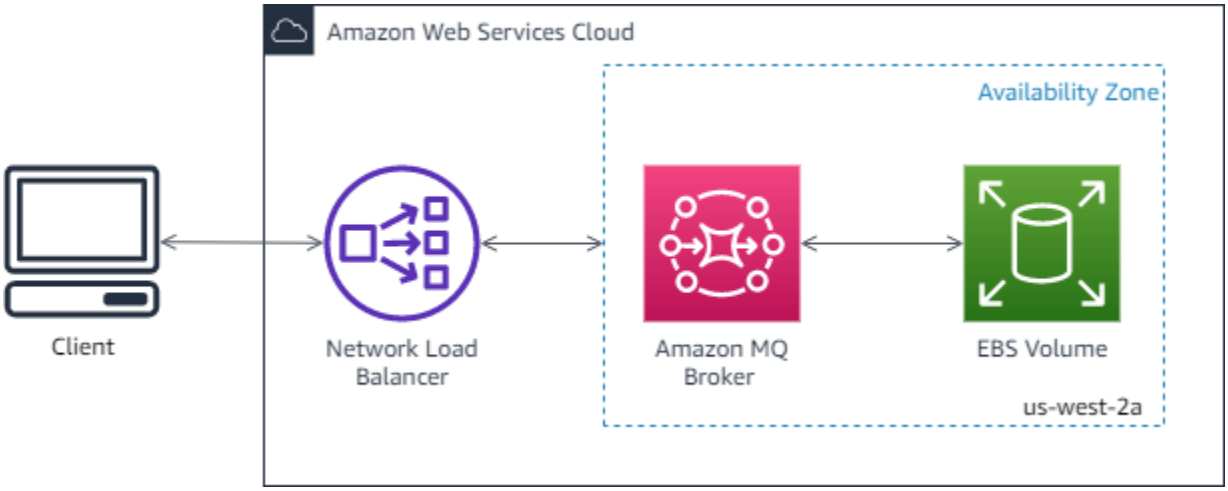
- Opsi 1: Amazon MQ untuk broker instans tunggal RabbitMQ
- Opsi 2: Amazon MQ untuk penyebaran cluster RabbitMQ

Opsi 1: Amazon MQ untuk broker instans tunggal RabbitMQ

Broker instans tunggal terdiri dari satu broker di satu Availability Zone di balik Penyeimbang Beban Jaringan (NLB). Broker berkomunikasi dengan aplikasi Anda dan dengan volume penyimpanan Amazon EBS. Amazon EBS menyediakan penyimpanan tingkat blok yang dioptimalkan untuk latensi rendah dan throughput tinggi.

Menggunakan Network Load Balancer memastikan bahwa titik akhir Amazon MQ untuk broker RabbitMQ Anda tetap tidak berubah jika instans broker diganti selama jendela pemeliharaan atau karena kegagalan perangkat keras Amazon yang mendasarinya. EC2 Penyeimbang Beban Jaringan memungkinkan aplikasi dan pengguna Anda untuk terus menggunakan titik akhir yang sama untuk terhubung ke broker.

Diagram berikut mengilustrasikan broker instans tunggan Amazon MQ for RabbitMQ.



Opsi 2: Amazon MQ untuk penyebaran cluster RabbitMQ

Deployment klaster adalah pengelompokan logis dari tiga node broker RabbitMQ di balik Penyeimbang Beban Jaringan, masing-masing membagikan pengguna, antrean, dan status terdistribusi di beberapa Availability Zone (AZ).

Dalam deployment klaster, Amazon MQ mengelola kebijakan broker secara otomatis untuk mengaktifkan pencerminan klasik di semua simpul, memastikan ketersediaan tinggi (HA). Setiap antrian yang dicerminkan terdiri dari satu simpul utama dan satu atau lebih cermin. Setiap antrean memiliki simpul utamanya sendiri. Semua operasi untuk antrean yang diberikan pertama-tama diterapkan pada simpul utama antrean lalu disebarkan ke cermin. Amazon MQ membuat kebijakan sistem default yang menetapkan `ha-mode` ke `all` dan `ha-sync-mode` ke `automatic`. Hal ini memastikan bahwa data direplikasi ke semua simpul dalam klaster di Availability Zone yang berbeda untuk daya tahan yang lebih baik.

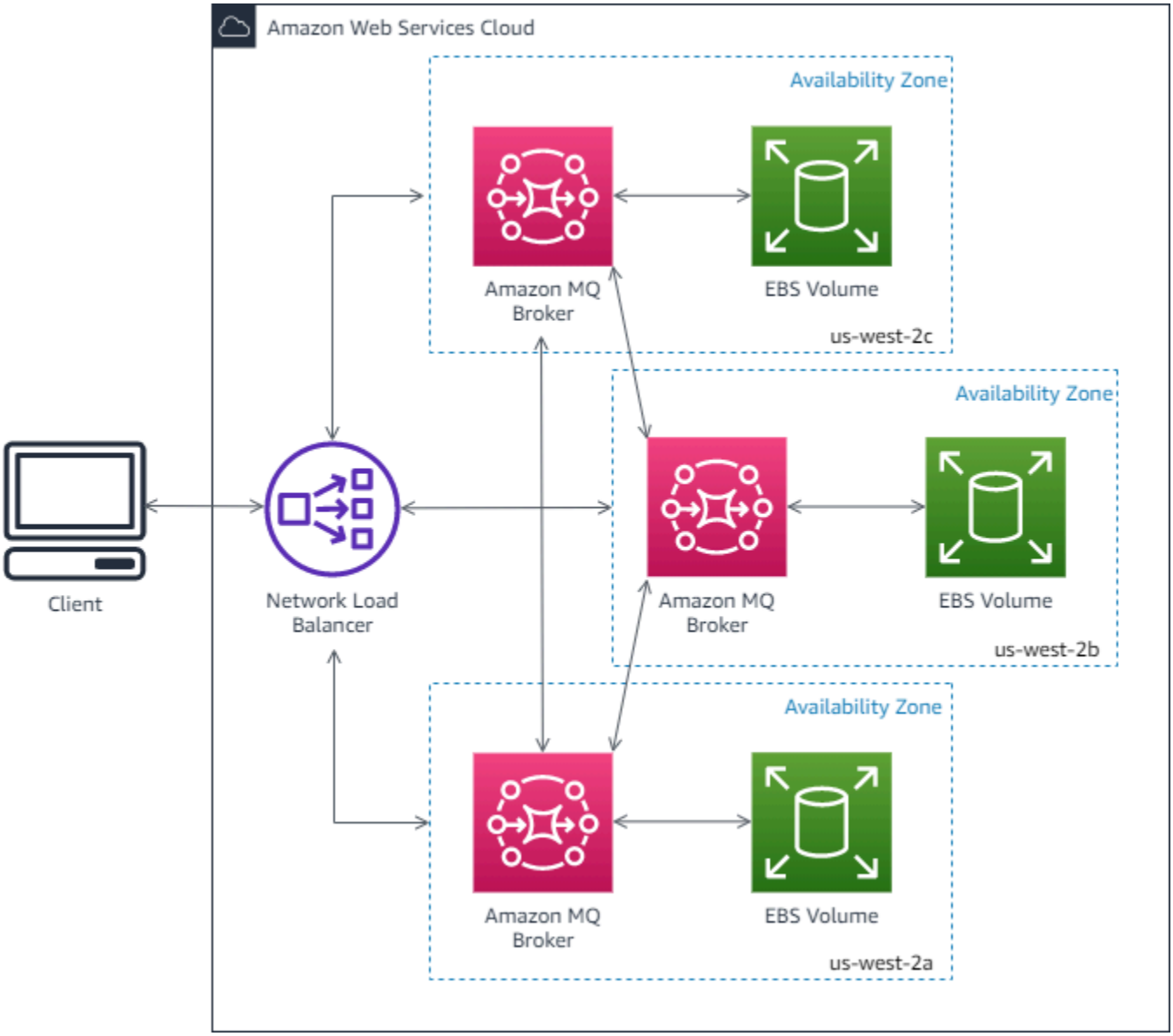
> **Note**
>
> Dalam penerapan klaster, jika terjadi pemadaman Availability Zone, Amazon MQ akan secara otomatis mencoba memindahkan node RabbitMQ yang terpengaruh ke AZ yang berbeda untuk mempertahankan ukuran cluster. Setelah pemadaman selesai, cluster akan secara otomatis diseimbangkan kembali di seluruh. AZs

> **Note**
>
> Selama jendela pemeliharaan, semua pemeliharaan ke klaster dilakukan satu simpul pada satu waktu, menjaga setidaknya dua simpul berjalan setiap saat. Setiap kali simpul

dinonaktifkan, koneksi klien ke simpul tersebut diputus dan perlu dibuat lagi. Anda harus memastikan bahwa kode klien dirancang untuk terhubung kembali secara otomatis ke klaster Anda. Untuk informasi selengkapnya tentang pemulihan koneksi, lihat the section called "Langkah 1: Secara otomatis pulih dari kegagalan jaringan".
Karena Amazon MQ menetapkan `ha-sync-mode: automatic`, selama jendela pemeliharaan, antrean akan menyinkronkan ketika setiap simpul kembali menggabungkan klaster. Sinkronisasi antrian memblokir semua operasi antrean lainnya. Anda dapat mengurangi dampak sinkronisasi antrean selama jendela pemeliharaan dengan membuat antrian tetap pendek.

Kebijakan default tidak boleh dihapus. Jika Anda menghapus kebijakan ini, Amazon MQ akan secara otomatis membuatnya kembali. Amazon MQ juga akan memastikan bahwa properti HA diterapkan ke semua kebijakan lain yang Anda buat pada broker terklaster. Jika Anda menambahkan kebijakan tanpa properti HA, Amazon MQ akan menambahkannya untuk Anda. Jika Anda menambahkan kebijakan dengan properti ketersediaan tinggi yang berbeda, Amazon MQ akan menggantinya. Untuk informasi selengkapnya tentang pencerminan klasik, lihat Antrian cermin klasik.

Diagram berikut mengilustrasikan deployment broker klaster RabbitMQ dengan tiga simpul di tiga Availability Zone (AZ), masing-masing dengan volume Amazon EBS sendiri dan status bersama. Amazon EBS menyediakan penyimpanan tingkat blok yang dioptimalkan untuk latensi rendah dan throughput tinggi.



Amazon MQ untuk jenis instans broker RabbitMQ

Deskripsi gabungan dari kelas instance broker (m7g) dan ukuran (besar, sedang) disebut tipe instance broker (misalnya, mq.m7g.large).

Sebaiknya gunakan tipe instans mq.m7g untuk penerapan cluster dan single instance.

Amazon MQ menyediakan setidaknya pemberitahuan 90 hari sebelum jenis instans mencapai akhir dukungan. Kami merekomendasikan untuk meningkatkan broker Anda ke jenis instans baru sebelum end-of-support tanggal untuk mencegah gangguan apa pun.

> **Important**
>
> Anda tidak dapat menurunkan versi broker dari tipe `mq.m7g` atau `mq.m5` instance ke tipe `mq.t3.micro` instans.
> Jenis `mq.t3.micro` instance tidak mendukung penerapan klaster.

Jenis instans untuk penerapan klaster m7g

Kami merekomendasikan penggunaan tipe `mq.m7g.x` instance dengan penerapan cluster. Tabel berikut menunjukkan jenis `mq.m7g.x` instance yang tersedia untuk penyebaran cluster.

Tipe Instans	vCPU	Memori (GiB)	Garis Dasar Jaringan/ Bandwidth Burst (Gbps)	Penggunaan yang disarankan	Penyimpanan	Ukuran volume disk per node (GB)
mq.m7g.medium	1	4	0,52/12,5	Evaluasi	EBS	5
mq.m7g.large	2	8	0,937/12,5	Produksi	EBS	15
mq.m7g.xlarge	4	16	1.876/12,5	Produksi	EBS	25
mq.m7g.2xlarge	8	32	3,75/15,0	Produksi	EBS	45
mq.m7g.4xlarge	16	64	7,5/15,0	Produksi	EBS	90
mq.m7g.8xlarge	32	128	15 Gigabit	Produksi	EBS	175

Tipe Instans	vCPU	Memori (GiB)	Garis Dasar Jaringan/ Bandwidth Burst (Gbps)	Penggunaan yang disarankan	Penyimpanan	Ukuran volume disk per node (GB)
mq.m7g.12xlarge	48	192	22,5 Gigabit	Produksi	EBS	260
mq.m7g.16xlarge	64	256	30 Gigabit	Produksi	EBS	345

Jenis instans untuk penerapan instans tunggal m7g

Tabel berikut menunjukkan jenis `mq.m7g.x` instance yang tersedia untuk penyebaran instance tunggal.

Tipe Instans	vCPU	Memori (GiB)	Garis Dasar Jaringan/ Bandwidth Burst (Gbps)	Penggunaan yang disarankan	Penyimpanan	Ukuran volume disk per node (GB)
mq.m7g.medium	1	4	0,52/12,5	Evaluasi	EBS	200
mq.m7g.large	2	8	0,937/12,5	Produksi	EBS	200
mq.m7g.xlarge	4	16	1.876/12,5	Produksi	EBS	200
mq.m7g.2xlarge	8	32	3,75/15,0	Produksi	EBS	200

Tipe Instans	vCPU	Memori (GiB)	Garis Dasar Jaringan/ Bandwidth Burst (Gbps)	Penggunaan yang disarankan	Penyimpanan	Ukuran volume disk per node (GB)
mq.m7g.4xlarge	16	64	7,5/15,0	Produksi	EBS	200
mq.m7g.8xlarge	32	128	15 Gigabit	Produksi	EBS	200
mq.m7g.12xlarge	48	192	22,5 Gigabit	Produksi	EBS	200
mq.m7g.16xlarge	64	256	39 Gigabit	Produksi	EBS	200

Jenis instans untuk penyebaran instance **`mq.m5`** tunggal

Tabel berikut menunjukkan jenis `mq.m5.x` instance yang tersedia untuk penyebaran instance tunggal

Tipe Instans	vCPU	Memori (GiB)	Garis Dasar Jaringan/ Bandwidth Burst (Gbps)	Penggunaan yang disarankan	Penyimpanan	Ukuran volume disk per node (GB)
mq.t3.mikro	2	1	0,064/5.0	Evaluasi	EBS	20
mq.m5.large	2	8	0,75/10.0	Produksi	EBS	200

Tipe Instans	vCPU	Memori (GiB)	Garis Dasar Jaringan/ Bandwidth Burst (Gbps)	Penggunaan yang disarankan	Penyimpanan	Ukuran volume disk per node (GB)
mq.m5.xlarge	4	16	1.25/10.0	Produksi	EBS	200
mq.m5.2xlarge	8	32	2.5/10.0	Produksi	EBS	200
mq.m5.4xlarge	16	64	5.0/10.0	Produksi	EBS	200

Jenis instans untuk penerapan **`mq.m5`** klaster

Tabel berikut menunjukkan jenis `mq.m5.x` instance yang tersedia untuk penyebaran cluster

Tipe Instans	vCPU	Memori (GiB)	Garis Dasar Jaringan/ Bandwidth Burst (Gbps)	Penggunaan yang disarankan	Penyimpanan	Ukuran volume disk per node (GB)
mq.m5.large	2	8	0,75/10.0	Produksi	EBS	200
mq.m5.xlarge	4	16	1.25/10.0	Produksi	EBS	200
mq.m5.2xlarge	8	32	2.5/10.0	Produksi	EBS	200

Tipe Instans	vCPU	Memori (GiB)	Garis Dasar Jaringan/ Bandwidth Burst (Gbps)	Penggunaan yang disarankan	Penyimpanan	Ukuran volume disk per node (GB)
mq.m5.4xlarge	16	64	5.0/10.0	Produksi	EBS	200

Amazon MQ untuk pedoman ukuran RabbitMQ

Anda dapat memilih jenis instans broker yang paling mendukung aplikasi Anda. Saat memilih jenis instans, pertimbangkan faktor-faktor yang akan memengaruhi kinerja broker:

- jumlah klien dan antrian
- volume pesan yang dikirim
- pesan disimpan dalam memori
- pesan berlebihan

Jenis instans broker `m7g.medium` yang lebih kecil direkomendasikan hanya untuk menguji kinerja aplikasi. Kami merekomendasikan jenis instans broker yang lebih besar `m7g.large` dan di atas atau tingkat produksi klien dan antrian, throughput tinggi, pesan dalam memori, dan pesan yang berlebihan.

> **Important**
>
> Anda tidak dapat menurunkan versi broker dari tipe `mq.m5` atau `mq.m7g` `mq.t3.micro` instance ke tipe instans.

Penting untuk menguji broker Anda untuk menentukan jenis dan ukuran instans yang sesuai untuk persyaratan pesan beban kerja Anda.

Selalu gunakan batas sumber daya default pada broker RabbitMQ 4 untuk menentukan ukuran instans yang sesuai untuk aplikasi Anda sesuai dengan praktik terbaik Amazon MQ. Batas sumber daya default ini didasarkan pada tipe tipe `m7g` instans dan antrian kuorum.

- Batas sumber daya default untuk penerapan instans tunggal m7g
- Batas sumber daya default untuk penerapan klaster m7g

Anda dapat meningkatkan nilai batas apa pun hingga nilai maksimum seperti yang ditentukan oleh jenis instans dan mode penerapan. Namun, kami sangat menyarankan Anda menguji kinerja broker dengan nilai yang meningkat sebelum digunakan dalam produksi.

- Batas sumber daya maksimum untuk penerapan instans tunggal m7g
- Batas sumber daya maksimum untuk penerapan klaster m7g
- Batas sumber daya maksimum untuk penerapan instans tunggal m5
- Batas sumber daya maksimum untuk penerapan cluster m5
- Pesan kesalahan

> **Note**
>
> Broker RabbitMQ 3.13 tidak datang dengan batas sumber daya default, tetapi kami sarankan Anda menggunakan default yang disarankan.

Batas sumber daya default

Amazon MQ untuk RabbitMQ mendukung konfigurasi batas sumber daya broker dari RabbitMQ 4 dan seterusnya. Saat Anda membuat broker, Amazon MQ secara otomatis menerapkan nilai default ke batas sumber daya ini. Default ini bertindak sebagai pagar pembatas untuk melindungi ketersediaan broker Anda sambil mengakomodasi pola penggunaan pelanggan umum. Anda dapat menyesuaikan perilaku broker Anda dengan mengubah nilai konfigurasi batas agar lebih sesuai dengan persyaratan beban kerja spesifik Anda.

Sebelum melakukan perubahan, harap dicatat:

Important

1. Perubahan konfigurasi dapat memengaruhi kinerja dan ketersediaan broker
2. Pahami dampaknya sebelum mengubah opsi konfigurasi default
3. Uji perubahan konfigurasi di lingkungan non-produksi terlebih dahulu
4. Pantau kesehatan broker setelah menerapkan perubahan

Important

Nilai default dan rentang yang didukung untuk konfigurasi ini bervariasi menurut versi RabbitMQ, tipe instans, dan mode penerapan broker.

Important

Catatan: Mengaitkan atau membuat broker dengan nilai konfigurasi di luar rentang yang didukung akan menghasilkan respons kesalahan.

Batas sumber daya default yang diterapkan untuk broker RabbitMQ 4.2 adalah

- Batas sumber daya default untuk penerapan instans tunggal m7g
- Batas sumber daya default untuk penerapan klaster m7g

Batas sumber daya default

Important

Amazon MQ untuk broker RabbitMQ 3, default dikonfigurasi dengan batas sumber daya maksimum dan Amazon MQ tidak memberikan kemampuan untuk mengganti konfigurasi batas sumber daya.

Nilai default untuk broker instans tunggal

Tipe instans	Koneksi per Node	Saluran per Node	Konsumer per saluran	Antrian	vhost	Sekop	Pertukaran	Ukuran pesan dalam Bytes
mq.m7g.medium	100	500	10	500	10	30	500	524288
mq.m7g.large	1.500	4,500	10	1.000	50	50	1.000	524288
mq.m7g.xlarge	3.000	9.000	10	2.000	100	100	2.000	524288
mq.m7g.2large	6.000	18.000	10	4.000	150	200	4.000	524288
mq.m7g.4large	12.000	36.000	10	8.000	200	400	8.000	524288
mq.m7g.8large	24.000	72.000	10	16.000	250	800	16.000	524288
mq.m7g.1xlarge	36.000	108.000	10	24.000	300	1.200	24.000	524288
mq.m7g.1xlarge	48.000	144.000	10	32.000	350	1.600	32.000	524288

Nilai default untuk broker klaster

Tipe instans	Koneksi per Node	Saluran per Node	Konsumer per saluran	Antrian	vhost	Sekop	Pertukaran	Ukuran pesan dalam Bytes
mq.m7g.n dium	100	300	10	100	10	10	100	524288
mq.m7g.la rge	500	1500	10	1.000	50	30	1.000	524288
mq.m7g.x arge	1000	3000	10	2.000	100	60	2.000	524288
mq.m7g.2 large	2000	6000	10	4.000	150	120	4.000	524288
mq.m7g.4 large	4000	12.000	10	8.000	200	240	8.000	524288
mq.m7g.8 large	8000	24.000	10	16.000	250	480	16.000	524288
mq.m7g.1 xlarge	12000	36.000	10	24.000	300	720	24000	524288
mq.m7g.1 xlarge	16.000	48.000	10	32.000	350	960	32.000	524288

Amazon MQ untuk batas sumber daya maksimum RabbitMQ

Pedoman ukuran untuk m7g dengan antrian kuorum untuk penerapan instans tunggal

Tabel berikut menunjukkan nilai batas maksimum untuk setiap jenis instans untuk broker instans tunggal.

Tipe Instans	Koneksi	Saluran	Konsumer per saluran	Antrian	Vhost	Sekop	Pertukaran	Ukuran Pesan dalam Byte
mq.m7g.medium	300	900	1.000	2.500	10	150	12500	134217728
mq.m7g.large	5.000	15.000	1.000	20.000	1500	250	100.000	134217728
mq.m7g.xlarge	10.000	30.000	1.000	30.000	1.500	500	150.000	134217728
mq.m7g.2large	20.000	60.000	1.000	40.000	1.500	1.000	200.000	134217728
mq.m7g.4large	40.000	120.000	1.000	60.000	1.500	2000	300.000	134217728
mq.m7g.8large	80.000	240.000	1.000	80.000	1.500	4000	400.000	134217728
mq.m7g.1xlarge	120.000	360.000	1.000	100.000	1.500	6.000	500.000	134217728
mq.m7g.1xlarge	160.000	480.000	1.000	120.000	1.500	8.000	600.000	134217728

Pedoman ukuran untuk m7g dengan antrian kuorum untuk penerapan klaster

Tabel berikut menunjukkan nilai batas maksimum untuk setiap jenis instans untuk broker cluster.

Tipe Instans	Koneksi per Node	Saluran per Node	Konsumer per saluran	Antrian	Vhost	Sekop	Pertukara n	Ukuran Pesan dalam Byte
mq.m7g.n dium	300	900	1.000	500	10	50	500	134217728
mq.m7g.la rge	5.000	15.000	1.000	10.000	1.500	150	50.000	134217728
mq.m7g.x arge	10.000	30.000	1.000	15.000	1.500	300	75.000	134217728
mq.m7g.2 large	20.000	60.000	1.000	20.000	1.500	600	100.000	134217728
mq.m7g.4 large	40.000	120.000	1.000	30.000	1.500	1200	150.000	134217728
mq.m7g.8 large	80.000	240.000	1.000	40.000	1.500	2,400	200.000	134217728
mq.m7g.1 xlarge	120.000	360.000	1.000	50.000	1.500	3.600	250.000	134217728
mq.m7g.1 xlarge	160.000	480.000	1.000	60.000	1.500	4,800	300.000	134217728

Batas sumber daya maksimum untuk penerapan instans tunggal M5

Tabel berikut menunjukkan nilai batas maksimum untuk setiap jenis instans untuk broker instans tunggal.

Tipe Instans	Koneksi	Saluran	Konsumen per saluran	Antrian	Vhost	Sekop
m5.large	5.000	15.000	1.000	30.000	1500	250

Tipe Instans	Koneksi	Saluran	Konsumen per saluran	Antrian	Vhost	Sekop
m5.xlarge	10.000	30.000	1.000	60.000	1500	500
m5.2xlarge	20.000	60.000	1.000	120.000	1500	1.000
m5.4xlarge	40.000	120.000	1000	240.000	1.000	2.000

Batas sumber daya maksimum untuk penerapan cluster m5

Tabel berikut menunjukkan nilai batas maksimum untuk setiap jenis instans untuk broker cluster.

Tipe Instans	Antrian	Konsumen per saluran	Sekop
m5.large	10.000	1.000	150
m5.xlarge	15.000	1.000	300
m5.2xlarge	20.000	1.000	600
m5.4xlarge	30.000	1.000	1200

Batas koneksi dan saluran berikut diterapkan per node:

Tipe Instans	Koneksi	Saluran
m5.large	5000	15.000
m5.xlarge	10.000	30.000
m5.2xlarge	20.000	60.000
m5.4xlarge	40.000	120.000

Nilai batas yang tepat untuk broker cluster mungkin lebih rendah dari nilai yang ditunjukkan tergantung pada jumlah node yang tersedia dan bagaimana RabbitMQ mendistribusikan sumber

daya di antara node yang tersedia. Jika Anda melebihi nilai batas, Anda dapat membuat koneksi baru ke node yang berbeda dan mencoba lagi, atau Anda dapat meningkatkan ukuran instance untuk meningkatkan batas maksimum

Pesan kesalahan

Pesan galat berikut dikembalikan ketika batas terlampaui. Semua nilai didasarkan pada batas instance **`m7.large`** tunggal.

> **Note**
>
> Kode kesalahan untuk pesan berikut dapat berubah berdasarkan pustaka klien yang Anda gunakan.

Koneksi

```
ConnectionClosedByBroker 500 "NOT_ALLOWED - connection refused: node
connection limit (5000) is reached"
```

Kanal

```
ConnectionClosedByBroker 1500 "NOT_ALLOWED - number of channels opened on
node 'rabbit@ip-10-0-23-173.us-west-2.compute.internal' has reached the
maximum allowed limit of (15,000)"
```

Konsumen

```
ConnectionClosedByBroker: (530, 'NOT_ALLOWED - reached maximum (1,000) of
consumers per channel')
```

Ukuran pesan maksimal

```
(406, 'PRECONDITION_FAILED - message size 524289 is larger than configured
max size 524288')
```

Pertukaran

```
(406, "PRECONDITION_FAILED - cannot declare exchange 'limit_test_3' in
vhost '/': exchange limit of 10 is reached")
```

> **Note**
>
> Pesan galat berikut menggunakan format HTTP Management API.

Antrian

```
{"error":"bad_request","reason":"cannot declare queue 'my_queue': queue
limit in cluster (10,000) is reached"}]
```

Sekop

```
{"error":"bad_request","reason":"Validation failed\n\ncomponent shovel is
limited to 150 per node\n"}
```

Vhost

```
{"error":"bad_request","reason":"cannot create vhost 'my_vhost': vhost
limit of 1500 is reached"}
```

Broker default Amazon MQ for RabbitMQ

Ketika Anda membuat broker Amazon MQ for RabbitMQ, Amazon MQ menerapkan serangkaian kebijakan broker default dan batas vhost untuk mengoptimalkan performa broker. Amazon MQ menerapkan batas vhost hanya untuk vhost default (`/`). Amazon MQ tidak akan menerapkan kebijakan default ke vhost yang baru dibuat. Kami merekomendasikan Anda menyimpan default ini untuk semua broker baru dan yang sudah ada. Namun, Anda dapat mengubah, menimpa, atau menghapus default ini kapan saja.

Amazon MQ menciptakan kebijakan broker dan batas vhost yang berbeda untuk Amazon MQ untuk RabbitMQ 3 dan RabbitMQ 4. Perbedaannya akan dibahas secara rinci dalam subbagian berikut.

Amazon MQ menciptakan kebijakan dan batas berdasarkan tipe instans dan mode deployment broker yang Anda pilih saat membuat broker. Kebijakan default diberi nama sesuai dengan mode deployment, seperti berikut:

Amazon MQ untuk RabbitMQ 3:

- Instans tunggal – `AWS-DEFAULT-POLICY-SINGLE-INSTANCE`
- Penerapan cluster — && `AWS-DEFAULT-POLICY-CLUSTER-MULTI-AZ` `AWS-DEFAULT-QUORUM-QUEUES-POLICY-CLUSTER-MULTI-AZ`

Amazon MQ untuk RabbitMQ 4:

- Instans tunggal – `AWS-DEFAULT-POLICY-SINGLE-INSTANCE`
- Penerapan cluster — && `AWS-DEFAULT-POLICY-CLUSTER` `AWS-DEFAULT-QUORUM-QUEUES-POLICY-CLUSTER-MULTI-AZ`

Untuk broker instans tunggal, Amazon MQ menetapkan nilai prioritas kebijakan ke `0`. Untuk menimpa nilai prioritas default, Anda dapat membuat kebijakan kustom Anda sendiri dengan nilai prioritas yang lebih tinggi. Untuk deployment klaster, Amazon MQ menetapkan nilai prioritas ke 1 untuk broker default. Untuk membuat kebijakan kustom Anda sendiri bagi klaster, tetapkan nilai prioritas yang lebih besar dari 1.

> **Note**
>
> Dalam deployment klaster, kebijakan broker `ha-mode` dan `ha-sync-mode` diperlukan untuk pencerminan klasik dan ketersediaan tinggi (HA). Pengaturan ini hanya berlaku untuk Amazon MQ untuk RabbitMQ 3 dan tidak dikonfigurasi untuk RabbitMQ 4.
> Jika Anda menghapus kebijakan default `AWS-DEFAULT-POLICY-CLUSTER-MULTI-AZ`, Amazon MQ menggunakan kebijakan `ha-all-AWS-OWNED-DO-NOT-DELETE` dengan nilai prioritas `0`. Hal ini memastikan bahwa kebijakan `ha-mode` dan `ha-sync-mode` yang diperlukan masih berlaku. Jika Anda membuat kebijakan kustom Anda sendiri, Amazon MQ secara otomatis menambahkan `ha-mode` dan `ha-sync-mode` ke definisi kebijakan Anda.

Topik

- Deskripsi kebijakan dan batas
- Nilai default yang direkomendasikan

Deskripsi kebijakan dan batas

Daftar berikut menjelaskan kebijakan default dan batas yang diterapkan Amazon MQ untuk broker yang baru dibuat. Nilai untuk `max-length`, `max-queues`, dan `max-connections` beragam menurut tipe instans dan mode deployment broker Anda. Nilai tersebut tercantum pada bagian Nilai default yang direkomendasikan.

Pengaturan pada broker RabbitMQ 3 dan RabbitMQ 4

- **`queue-mode: lazy`** (kebijakan) - Memungkinkan antrean malas. Secara default, antrean menyimpan cache dalam memori dari pesan, memungkinkan broker mengirimkan pesan kepada konsumen secepat mungkin. Hal ini dapat menyebabkan broker kehabisan memori dan memicu alarm memori tinggi. Antrean malas mencoba untuk memindahkan pesan ke disk sedini mungkin. Ini berarti bahwa lebih sedikit pesan disimpan dalam memori dalam kondisi operasi normal. Menggunakan antrean malas, Amazon MQ for RabbitMQ dapat mendukung beban olahpesan yang jauh lebih besar dan antrean yang lebih panjang. Perhatikan bahwa untuk kasus penggunaan tertentu, broker dengan antrean malas mungkin memiliki performa yang sedikit lebih lambat. Ini karena pesan dipindahkan dari disk ke broker, berlawanan dengan mengirim pesan dari cache dalam memori.

  > Mode deployment
  >
  > Instans tunggal, klaster

- **`max-length: number-of-messages`** (klaster) - Menetapkan batas jumlah pesan dalam antrean. Pada deployment klaster, batas mencegah jeda sinkronisasi antrean dalam kasus seperti boot ulang broker, atau mengikuti jendela pemeliharaan.

  > Mode deployment
  >
  > Klaster

- **`overflow: reject-publish`** (kebijakan) — Menegakkan kebijakan `max-length` ke antrean untuk menolak pesan baru setelah jumlah pesan dalam antrean mencapai nilai `max-length`. Untuk memastikan bahwa pesan tidak hilang jika antrian dalam keadaan berlebih, aplikasi klien yang memublikasikan pesan ke broker harus menerapkan konfirmasi penerbit. Untuk informasi tentang mengimplementasikan konfirmasi penerbit, lihat konfirmasi Penerbit di situs web RabbitMQ.

  > Mode deployment
  >
  > Kluster

Pengaturan khusus untuk RabbitMQ 3

- **`max-queues: number-of-queues-per-vhost`** (batas vhost) - Mengatur batas untuk jumlah antrean dalam broker. Mirip dengan definisi kebijakan `max-length`, membatasi jumlah antrean dalam deployment klaster mencegah jeda sinkronisasi antrean setelah boot ulang broker atau jendela pemeliharaan. Membatasi antrean juga mencegah jumlah penggunaan CPU yang berlebihan untuk mempertahankan antrean.

  > Mode deployment
  >
  > Instans tunggal, klaster

- **`max-connections: number-of-connections-per-vhost`** (batas vhost) - Mengatur batas untuk jumlah koneksi klien ke broker. Membatasi jumlah koneksi sesuai dengan nilai yang disarankan mencegah penggunaan memori broker yang berlebihan, yang dapat mengakibatkan broker meningkatkan alarm memori tinggi dan menghentikan operasi.

  > Mode deployment
  >
  > Instans tunggal, klaster

Nilai default yang direkomendasikan

> Important
>
> `max-queues`dan hanya `max-connections` diterapkan ke Amazon MQ untuk RabbitMQ 3.

> Note
>
> Batas default `max-length` dan `max-queue` diuji dan dievaluasi berdasarkan ukuran pesan rata-rata sebesar 5 kB. Jika ukuran pesan jauh lebih besar dari 5 kB, Anda harus menyesuaikan dan mengurangi batas `max-length` dan `max-queue`.

Tabel berikut mencantumkan nilai batas default untuk broker yang baru dibuat. Amazon MQ menerapkan nilai tersebut sesuai dengan tipe instans broker dan mode deployment.

Tipe instans	Mode deployment	**max-length**	**max-queues**	**max-connections**
mq.m7g.medium	Instans tunggal	N/A	2.500	100
	Kluster	500.000	100	100
mq.m7g.large	Instans tunggal	N/A	20.000	5.000
	Kluster	8.000.000	10.000	5.000
mq.m7g.xlarge	Instans tunggal	N/A	30.000	10.000
	Kluster	9.000.000	15.000	10.000
mq.m7g.2xlarge	Instans tunggal	N/A	40.000	20.000
	Kluster	10.000.000	40.000	20.000
mq.m7g.4xlarge	Instans tunggal	N/A	60.000	40.000
	Kluster	12.000.000	30.000	40.000
mq.m7g.8xlarge	Instans tunggal	N/A	80.000	80.000
	Kluster	20.000.000	40.000	80.000
mq.m7g.12xlarge	Instans tunggal	N/A	100.000	120.000
	Kluster	30.000.000	20.000	120.000
mq.m7g.16xlarge	Instans tunggal	N/A	120.000	160.000
	Kluster	40.000.000	50.000	160.000

Tipe instans	Mode deployment	**max-length**	**max-queues**	**max-connections**
t3.micro	Instans tunggal	N/A	500	500

Tipe instans	Mode deployment	`max-length`	`max-queues`	`max-connections`
m5.large	Instans tunggal	N/A	20.000	4.000
m5.large	Klaster	8.000.000	10.000	15.000
m5.xlarge	Instans tunggal	N/A	30.000	8.000
m5.xlarge	Klaster	9.000.000	10.000	20.000
m5.2xlarge	Instans tunggal	N/A	60.000	15.000
m5.2xlarge	Klaster	10.000.000	10.000	40.000
m5.4xlarge	Instans tunggal	N/A	150.000	30.000
m5.4xlarge	Klaster	12.000.000	10.000	100.000

Mengkonfigurasi broker RabbitMQ

Konfigurasi berisi semua pengaturan untuk broker RabbitMQ Anda dalam format Sotong. Anda dapat membuat konfigurasi sebelum membuat broker. Kemudian Anda dapat menerapkan konfigurasi ke satu atau lebih broker.

Atribut

Konfigurasi broker memiliki beberapa atribut, misalnya:

- Nama (MyConfiguration)
- Sebuah ID (c-1234a5b6-78cd-901e-2fgh-3i45j6k178l9)
- Nama Sumber Daya Amazon (ARN) (arn:aws:mq:us-east-2:123456789012:configuration:c-1234a5b678cd-901e-2fgh-3i45j6k178l9)

Untuk daftar lengkap atribut konfigurasi, lihat di Referensi REST API Amazon MQ:

- ID Operasi REST: Konfigurasi
- ID Operasi REST: Konfigurasi

Untuk daftar lengkap atribut revisi konfigurasi, lihat hal berikut:

- ID Operasi REST: Revisi Konfigurasi
- ID Operasi REST: Revisi Konfigurasi

Topik

- Membuat dan menerapkan konfigurasi broker RabbitMQ
- Edit Amazon MQ untuk Revisi Konfigurasi RabbitMQ
- Nilai yang dapat dikonfigurasi untuk RabbitMQ di Amazon MQ
- Dukungan ARN dalam konfigurasi RabbitMQ

Membuat dan menerapkan konfigurasi broker RabbitMQ

Konfigurasi berisi semua pengaturan untuk broker RabbitMQ Anda dalam format Sotong. Anda dapat membuat konfigurasi sebelum membuat broker. Anda kemudian dapat menerapkan konfigurasi ke satu atau lebih broker

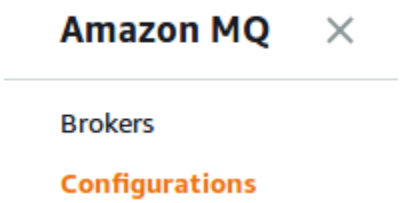
Contoh berikut menunjukkan bagaimana Anda dapat membuat dan menerapkan konfigurasi broker RabbitMQ menggunakan. Konsol Manajemen AWS

> **Important**
>
> Anda hanya dapat menghapus konfigurasi menggunakan `DeleteConfiguration` API. Untuk informasi selengkapnya, lihat Konfigurasi di Referensi API Amazon MQ.

Buat Konfigurasi Baru

Untuk menerapkan konfigurasi ke broker Anda, Anda harus terlebih dahulu membuat konfigurasi.

1. Masuk ke konsol Amazon MQ.
2. Di sebelah kiri, perluas panel navigasi dan pilih Konfigurasi.



3. Di halaman Konfigurasi, pilih Buat konfigurasi.
4. Di halaman Buat konfigurasi, pada bagian Detail, ketik Nama konfigurasi (Misalnya, `MyConfiguration`) dan pilih versi Mesin broker.

   Untuk mempelajari lebih lanjut tentang versi mesin RabbitMQ yang didukung oleh Amazon MQ untuk RabbitMQ, lihat. the section called “Manajemen versi”
5. Pilih Buat konfigurasi.

Buat Revisi Konfigurasi Baru

Setelah Anda membuat konfigurasi, Anda harus mengedit konfigurasi menggunakan revisi konfigurasi.

1. Dari daftar konfigurasi, pilih ***`MyConfiguration`***.

   > **Note**
   >
   > Revisi konfigurasi pertama selalu dibuat untuk Anda ketika Amazon MQ membuat konfigurasi.

   Pada ***`MyConfiguration`***halaman, jenis dan versi mesin broker yang digunakan revisi konfigurasi baru Anda (misalnya, RabbitMQ 3.xx.xx) ditampilkan.
2. Pada tab Detail konfigurasi, nomor revisi konfigurasi, deskripsi, dan konfigurasi broker dalam format Cuttlefish ditampilkan.

   > **Note**
   >
   > Mengedit konfigurasi saat ini membuat revisi konfigurasi baru.
3. Pilih Edit konfigurasi dan buat perubahan pada konfigurasi Cuttlefish.
4. Pilih Simpan.

Kotak dialog Simpan revisi akan ditampilkan.

5. (Opsional) Tipe `A description of the changes in this revision.`
6. Pilih Simpan.

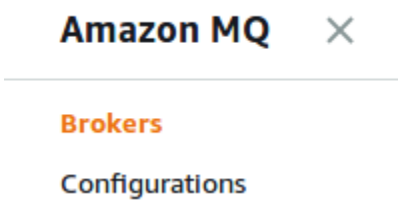
   Revisi konfigurasi baru akan disimpan.

> **Important**
>
> Pembuatan perubahan pada konfigurasi tidak akan segera menerapkan perubahan ke broker. Untuk menerapkan perubahan Anda, Anda harus menunggu jendela pemeliharaan berikutnya atau reboot broker.
> Saat ini, Anda tidak dapat menghapus konfigurasi.

Terapkan Revisi Konfigurasi ke Broker Anda

Setelah membuat revisi konfigurasi, Anda dapat menerapkan revisi konfigurasi ke broker Anda.

1. Di sebelah kiri, perluas panel navigasi dan pilih Broker.



2. Dari daftar broker, pilih broker Anda (misalnya, MyBroker) dan kemudian pilih Edit.
3. Pada *`MyBroker`* halaman Edit, di bagian Konfigurasi, pilih Konfigurasi dan Revisi dan kemudian pilih Jadwal Modifikasi.
4. Di bagian Jadwalkan perubahan broker, pilih apakah akan menerapkan perubahan Selama jendela pemeliharaan terjadwal berikutnya atau Segera.

> **Important**
>
> Pialang instans tunggal sedang offline saat di-boot ulang. Untuk broker cluster, hanya satu node yang turun pada satu waktu sementara broker melakukan reboot.

5. Pilih Terapkan.

Revisi konfigurasi Anda diterapkan ke broker pada waktu yang ditentukan.

Edit Amazon MQ untuk Revisi Konfigurasi RabbitMQ

Petunjuk berikut menjelaskan cara mengedit revisi konfigurasi untuk broker Anda.

1. Masuk ke konsol Amazon MQ.
2. Dari daftar broker, pilih broker Anda (misalnya, MyBroker) dan kemudian pilih Edit.
3. Pada ***MyBroker***halaman, pilih Edit.
4. Pada *MyBroker* halaman Edit, di bagian Konfigurasi, pilih Konfigurasi dan Revisi dan kemudian pilih Edit.

   > **Note**
   >
   > Kecuali Anda memilih konfigurasi ketika membuat broker, revisi konfigurasi pertama selalu dibuat untuk Anda ketika Amazon MQ membuat broker.

   Pada ***MyBroker***halaman, jenis dan versi mesin broker yang digunakan konfigurasi (misalnya, RabbitMQ 3.xx.xx) ditampilkan.
5. Pada tab Detail konfigurasi, nomor revisi konfigurasi, deskripsi, dan konfigurasi broker dalam format Cuttlefish ditampilkan.

   > **Note**
   >
   > Mengedit konfigurasi saat ini membuat revisi konfigurasi baru.
6. Pilih Edit konfigurasi dan buat perubahan pada konfigurasi Cuttlefish.
7. Pilih Simpan.

   Kotak dialog Simpan revisi akan ditampilkan.
8. (Opsional) Tipe `A description of the changes in this revision.`
9. Pilih Simpan.

   Revisi konfigurasi baru akan disimpan.

> **Important**
>
> Pembuatan perubahan pada konfigurasi tidak akan segera menerapkan perubahan ke broker. Untuk menerapkan perubahan Anda, Anda harus menunggu jendela pemeliharaan berikutnya atau reboot broker.
> Saat ini, Anda tidak dapat menghapus konfigurasi.

Nilai yang dapat dikonfigurasi

Anda dapat mengatur nilai opsi konfigurasi broker berikut dengan memodifikasi file konfigurasi broker di Konsol Manajemen AWS.

Selain nilai yang dijelaskan dalam tabel berikut, Amazon MQ mendukung opsi konfigurasi broker tambahan yang terkait dengan otentikasi dan otorisasi serta batas sumber daya. Untuk informasi selengkapnya tentang opsi konfigurasi ini, lihat

- OAuth 2.0 konfigurasi
- Konfigurasi LDAP
- Konfigurasi HTTP
- Konfigurasi SSL
- Konfigurasi mTLS
- Dukungan ARN
- Batas sumber daya
- Konfigurasi SSL klien AMQP

Konfigurasi	nilai default	Nilai yang Direkomendasikan	Nilai	Versi yang Berlaku	Deskripsi
consumer_timeout	1800000 ms (30 menit)	1800000 ms (30 menit)	0 hingga 2.147.483.647 ms. Amazon MQ juga	Semua versi	Batas waktu pada pengakuan pengiriman konsumen

Konfigurasi	nilai default	Nilai yang Direkomendasikan	Nilai	Versi yang Berlaku	Deskripsi
			mendukung nilai 0, yang berarti “tak terbatas”.		untuk mendeteksi kapan konsumen tidak melakukan pengiriman.
detak jantung	60 detik	60 detik	60 hingga 3600 detik	Semua versi	Mendefinisikan waktu sebelum koneksi dianggap tidak tersedia oleh RabbitMQ.

Konfigurasi	nilai default	Nilai yang Direkomendasikan	Nilai	Versi yang Berlaku	Deskripsi
management.restrictions.operator_policy_changes.disabled	true	true	benar, salah	Semua versi	Menonaktifkan membuat perubahan pada kebijakan operator. Jika Anda membuat perubahan ini, Anda sangat dianjurkan untuk memasukkan properti HA dalam kebijakan operator Anda sendiri.
quorum_queue.property_equivalence.relaxed_checks_on_redeclaration	true	true	benar, salah	Semua versi	Saat disetel ke TRUE, aplikasi Anda menghindari pengecualian channel saat mendeklarasikan ulang antrian kuorum.

Konfigurasi	nilai default	Nilai yang Direkomendasikan	Nilai	Versi yang Berlaku	Deskripsi
secure.management.http.headers.enabled	true	true	benar, salah	Semua versi	Mengaktifkan header keamanan HTTP yang tidak dapat dimodifikasi.

Mengkonfigurasi pengakuan pengiriman konsumen

Anda dapat mengonfigurasi consumer_timeout untuk mendeteksi saat konsumen tidak melakukan pengiriman. Jika konsumen tidak mengirimkan pengakuan dalam nilai batas waktu, saluran akan ditutup. Misalnya, jika Anda menggunakan nilai default 1800000 milidetik, jika konsumen tidak mengirim pemberitahuan pengiriman dalam 1800000 milidetik, saluran akan ditutup. Amazon MQ juga mendukung nilai 0, yang berarti “tak terbatas”.

Mengkonfigurasi detak jantung

Anda dapat mengonfigurasi batas waktu detak jantung untuk mengetahui kapan koneksi terganggu atau gagal. Nilai detak jantung menentukan batas waktu sebelum koneksi dianggap turun.

Mengkonfigurasi kebijakan operator

Kebijakan operator default pada setiap host virtual memiliki properti HA yang direkomendasikan berikut:

```
{
"name": "default_operator_policy_AWS_managed",
"pattern": ".*",
"apply-to": "all",
"priority": 0,
"definition": {
"ha-mode": "all",
"ha-sync-mode": "automatic"
}
}
```

Perubahan kebijakan operator melalui Konsol Manajemen AWS atau Management API tidak tersedia secara default. Anda dapat mengaktifkan perubahan dengan menambahkan baris berikut ke konfigurasi broker:

```
management.restrictions.operator_policy_changes.disabled=false
```

Jika Anda membuat perubahan ini, Anda sangat dianjurkan untuk memasukkan properti HA dalam kebijakan operator Anda sendiri.

Mengkonfigurasi pemeriksaan santai pada deklarasi antrian

Jika Anda telah memigrasikan antrian klasik ke antrian kuorum tetapi tidak memperbarui kode klien, Anda dapat menghindari pengecualian saluran saat mendeklarasikan ulang antrian kuorum dengan mengonfigurasi quorum_queue.property_equivalence.relaxed_checks_on_redeclaration disetel ke true.

Mengkonfigurasi header keamanan HTTP

Konfigurasi secure.management.http.headers.enabled memungkinkan header keamanan HTTP berikut:

- X-Content-Type-Options: nosniff: mencegah browser melakukan sniffing konten, algoritma yang digunakan untuk menyimpulkan format file situs web.
- X-Frame-Options: DENY: mencegah orang lain menyematkan plugin manajemen ke dalam bingkai di situs web mereka sendiri untuk menipu orang lain
- Strict-Transport-Security: max-age=47304000; includeSubDomains: memaksa browser untuk menggunakan HTTPS saat membuat koneksi berikutnya ke situs web dan subdomainnya untuk jangka waktu yang lama (1,5 tahun).

Amazon MQ untuk broker RabbitMQ yang dibuat pada versi 3.10 dan di atasnya akan memiliki secure.management.http.headers.enabled disetel ke true secara default. Anda dapat mengaktifkan header keamanan HTTP ini dengan menyetel secure.management.http.headers.enabled ke true. Jika Anda ingin memilih keluar dari header keamanan HTTP ini, setel secure.management.http.headers.enabled ke false.

Mengkonfigurasi otentikasi dan otorisasi OAuth 2.0

Untuk informasi tentang opsi konfigurasi OAuth 2.0 dan menyiapkan otentikasi OAuth 2.0 untuk broker Anda, lihat Konfigurasi OAuth 2.0 yang didukung dan Menggunakan otentikasi dan otorisasi OAuth 2.0.

Mengkonfigurasi otentikasi dan otorisasi LDAP

Untuk informasi tentang opsi konfigurasi LDAP dan menyiapkan otentikasi LDAP untuk broker Anda, lihat Konfigurasi LDAP yang didukung dan. Menggunakan otentikasi dan otorisasi LDAP

Mengkonfigurasi otentikasi dan otorisasi HTTP

Untuk informasi tentang nilai konfigurasi otentikasi HTTP dan menyiapkan otentikasi HTTP untuk broker Anda, lihat otentikasi dan otorisasi HTTP dan. Menggunakan otentikasi dan otorisasi HTTP

> **Note**
>
> Plugin otentikasi HTTP hanya tersedia untuk Amazon MQ untuk RabbitMQ versi 4 ke atas.

Mengkonfigurasi otentikasi sertifikat SSL

Untuk informasi tentang nilai konfigurasi otentikasi sertifikat SSL dan menyiapkan otentikasi sertifikat SSL untuk broker Anda, lihat otentikasi sertifikat SSL dan. Menggunakan otentikasi sertifikat SSL

> **Note**
>
> Plugin otentikasi sertifikat SSL hanya tersedia untuk Amazon MQ untuk RabbitMQ versi 4 ke atas.

Mengkonfigurasi mTL

Amazon MQ untuk RabbitMQ mendukung TLS bersama (MTLS) untuk koneksi aman ke berbagai titik akhir dan layanan eksternal. mTLS menyediakan keamanan yang ditingkatkan dengan mengharuskan klien dan server untuk mengautentikasi menggunakan sertifikat.

> **Note**
>
> Penggunaan otoritas sertifikat swasta untuk mTLS hanya tersedia untuk Amazon MQ untuk RabbitMQ versi 4 ke atas.

> **Important**
>
> Amazon MQ untuk RabbitMQ memberlakukan penggunaan AWS ARNs untuk sertifikat dan file kunci pribadi. Lihat dukungan ARN dalam konfigurasi RabbitMQ untuk detail selengkapnya.

Di halaman ini

- Titik akhir AMQP
- Plugin manajemen RabbitMQ
- Plugin RabbitMQ 2.0 OAuth
- Plugin otentikasi HTTP RabbitMQ
- Plugin RabbitMQ LDAP
- Koneksi klien AMQP

Titik akhir AMQP

Konfigurasikan mTL untuk koneksi klien ke titik akhir AMQP. Ini digunakan dengan otentikasi sertifikat SSL. Untuk konfigurasi yang didukung, lihatOtentikasi sertifikat SSL.

Plugin manajemen RabbitMQ

Konfigurasikan mTLS untuk koneksi ke antarmuka manajemen RabbitMQ.

> **Note**
>
> MTL yang ketat tidak didukung untuk API manajemen.

Konfigurasi yang didukung

`aws.arns.management.ssl.cacertfile`

File otoritas sertifikat untuk memvalidasi sertifikat klien yang terhubung ke antarmuka manajemen.

`management.ssl.verify`

Mode verifikasi rekan. Nilai yang didukung:`verify_none`, `verify_peer`

`management.ssl.depth`

Kedalaman rantai sertifikat maksimum untuk verifikasi.

`management.ssl.hostname_verification`

Mode verifikasi nama host. Nilai yang didukung:`wildcard`, `none`

Opsi SSL yang tidak didukung

Nilai konfigurasi SSL berikut tidak didukung:

Lihat daftar lengkap

- `management.ssl.cert`
- `management.ssl.client_renegotiation`
- `management.ssl.dh`
- `management.ssl.dhfile`
- `management.ssl.fail_if_no_peer_cert`
- `management.ssl.honor_cipher_order`
- `management.ssl.honor_ecc_order`
- `management.ssl.key.RSAPrivateKey`
- `management.ssl.key.DSAPrivateKey`
- `management.ssl.key.PrivateKeyInfo`
- `management.ssl.log_alert`
- `management.ssl.password`
- `management.ssl.psk_identity`
- `management.ssl.reuse_sessions`
- `management.ssl.secure_renegotiate`

- `management.ssl.versions.$version`
- `management.ssl.sni`

Plugin RabbitMQ 2.0 OAuth

Konfigurasikan mTL untuk koneksi dari Amazon MQ ke penyedia identitas OAuth 2.0. Untuk konfigurasi yang didukung, lihatOAuth 2.0 otentikasi dan otorisasi.

Plugin otentikasi HTTP RabbitMQ

Konfigurasikan mTL untuk koneksi dari Amazon MQ ke server otentikasi HTTP. Untuk konfigurasi yang didukung, lihatOtentikasi dan otorisasi HTTP.

Plugin RabbitMQ LDAP

Konfigurasikan mTL untuk koneksi dari Amazon MQ ke server LDAP. Untuk konfigurasi yang didukung, lihatOtentikasi dan otorisasi LDAP.

Koneksi klien AMQP

Konfigurasikan verifikasi rekan TLS untuk koneksi klien AMQP yang digunakan oleh federasi dan sekop. Untuk informasi selengkapnya, lihat konfigurasi SSL klien AMQP.

> **Important**
>
> Amazon MQ saat ini tidak mendukung konfigurasi sertifikat klien untuk koneksi klien AMQP. Akibatnya, federasi dan sekop tidak dapat terhubung ke broker berkemampuan MTLS yang memerlukan otentikasi sertifikat klien.

Konfigurasi Batas Sumber Daya

Amazon MQ untuk RabbitMQ mendukung konfigurasi batas sumber daya broker dari RabbitMQ 4 dan seterusnya. Saat Anda membuat broker, Amazon MQ secara otomatis menerapkan nilai default ke batas sumber daya ini. Default ini bertindak sebagai pagar pembatas untuk melindungi ketersediaan broker Anda sambil mengakomodasi pola penggunaan pelanggan umum. Anda dapat menyesuaikan perilaku broker Anda dengan mengubah nilai konfigurasi batas agar lebih sesuai dengan persyaratan beban kerja spesifik Anda. Untuk detail selengkapnya tentang nilai default dan maksimum yang diizinkan, lihatthe section called “Pedoman ukuran”.

Nama sumber daya dan kunci konfigurasi

Nama Sumber Daya	Kunci Konfigurasi
Koneksi	`connection_max`
Channel	`channel_max_per_node`
Antrean	`cluster_queue_limit`
Vhost	`vhost_max`
Sekop	`runtime_parameters.limits.shovel`
.exchange	`cluster_exchange_limit`
Konsumen per saluran	`consumer_max_per_channel`
Ukuran pesan maksimal	`max_message_size`

Cara mengganti batas sumber daya

Anda dapat mengganti batas sumber daya menggunakan Amazon MQ API dan konsol Amazon MQ.

Contoh berikut menunjukkan cara mengganti batas default hitungan antrian menggunakan: AWS CLI

```
aws mq update-configuration --configuration-id <config-id> --data "$(echo
 "cluster_queue_limit=500" | base64 --wrap=0)"
```

Pemanggilan yang berhasil membuat revisi konfigurasi. Anda harus mengaitkan konfigurasi ke broker RabbitMQ Anda dan me-reboot broker untuk menerapkan penggantian. Untuk lebih jelasnya lihat RabbitMQ Broker Configurations

Kesalahan penggantian batas sumber daya

Mengaitkan atau membuat broker dengan nilai konfigurasi di luar rentang yang didukung menghasilkan respons kesalahan yang mirip dengan berikut ini:

```
Configuration Revision N for configuration:cluster_queue_limit has limit: of value:
 100000000 larger than maximum allowed limit:5000
```

Dukungan ARN dalam konfigurasi RabbitMQ

Amazon MQ untuk RabbitMQ mendukung AWS ARNs nilai beberapa pengaturan konfigurasi RabbitMQ. Ini diaktifkan oleh plugin komunitas RabbitMQ rabbitmq-aws. Plugin ini dikembangkan dan dikelola oleh Amazon MQ dan juga dapat digunakan di broker RabbitMQ yang dihosting sendiri yang tidak dikelola oleh Amazon MQ.

⚠ Pertimbangan penting

- Nilai ARN yang diselesaikan yang diambil oleh plugin aws diteruskan langsung ke proses RabbitMQ saat runtime. Mereka tidak disimpan di tempat lain di node RabbitMQ.
- Amazon MQ untuk RabbitMQ memerlukan peran IAM yang dapat diasumsikan oleh Amazon MQ untuk mengakses yang dikonfigurasi. ARNs Ini dikonfigurasi dengan pengaturan`aws.arns.assume_role_arn`.
- Pengguna yang menelepon CreateBroker atau UpdateBroker APIs dengan konfigurasi broker yang menyertakan peran IAM harus memiliki `iam:PassRole` izin untuk peran itu.
- Peran IAM harus ada di AWS akun yang sama dengan broker RabbitMQ. Semua ARNs dalam konfigurasi harus ada di AWS wilayah yang sama dengan broker RabbitMQ.
- Amazon MQ menambahkan kunci bersyarat global IAM `aws:SourceAccount` dan `aws:SourceArn` saat mengasumsikan peran IAM. Nilai-nilai ini harus digunakan dalam kebijakan IAM yang melekat pada peran untuk perlindungan wakil yang membingungkan.

Di halaman ini

- Kunci yang didukung
- Sampel kebijakan IAM
- Validasi akses
- Negara karantina broker terkait
- Contoh skenario

Kunci yang didukung

Peran IAM yang diperlukan

`aws.arns.assume_role_arn`

IAM berperan ARN yang diasumsikan Amazon MQ untuk mengakses sumber daya lain. AWS Diperlukan ketika konfigurasi ARN lainnya digunakan.

Titik akhir AMQP

Kunci konfigurasi	Deskripsi
`aws.arns.ssl_options.cacertfile`	File otoritas sertifikat untuk koneksi SSL/TLS klien. Amazon MQ memerlukan penggunaan Amazon S3 atau untuk menyimpan sertifikat.

Plugin manajemen RabbitMQ

Kunci konfigurasi	Deskripsi
`aws.arns.management.ssl.cacertfile`	File otoritas sertifikat untuk SSL/TLS koneksi antarmuka manajemen. Amazon MQ memerlukan penggunaan Amazon S3 atau untuk menyimpan sertifikat.

Plugin RabbitMQ 2.0 OAuth

Kunci konfigurasi	Deskripsi
`aws.arns.auth_oauth2.https.cacertfile`	File otoritas sertifikat untuk koneksi OAuth 2.0 HTTPS. Amazon MQ memerlukan penggunaan Amazon S3 atau untuk menyimpan sertifikat.

Plugin otentikasi HTTP RabbitMQ

Kunci konfigurasi	Deskripsi
`aws.arns.auth_http.ssl_options.cacertfile`	File otoritas sertifikat untuk SSL/TLS koneksi otentikasi HTTP. Amazon MQ memerlukan penggunaan Amazon S3 atau untuk menyimpan sertifikat.
`aws.arns.auth_http.ssl_options.certfile`	File sertifikat untuk koneksi TLS timbal balik antara Amazon MQ dan server otentikasi HTTP. Amazon MQ memerlukan penggunaan Amazon S3 atau untuk menyimpan sertifikat.
`aws.arns.auth_http.ssl_options.keyfile`	File kunci pribadi untuk koneksi TLS timbal balik antara Amazon MQ dan server otentikasi HTTP. Amazon MQ membutuhkan penggunaan AWS Secrets Manager untuk menyimpan kunci pribadi.

Plugin RabbitMQ LDAP

Kunci konfigurasi	Deskripsi
`aws.arns.auth_ldap.ssl_options.cacertfile`	File otoritas sertifikat untuk SSL/TLS koneksi LDAP. Amazon MQ memerlukan penggunaan Amazon S3 atau untuk menyimpan sertifikat.
`aws.arns.auth_ldap.ssl_options.certfile`	File sertifikat untuk koneksi TLS timbal balik antara Amazon MQ dan server LDAP. Amazon MQ memerlukan penggunaan Amazon S3 atau untuk menyimpan sertifikat.
`aws.arns.auth_ldap.ssl_options.keyfile`	File kunci pribadi untuk koneksi TLS timbal balik antara Amazon MQ dan server LDAP. Amazon MQ membutuhkan penggunaan AWS Secrets Manager untuk menyimpan kunci pribadi.
`aws.arns.auth_ldap.dn_lookup_bind.password`	Kata sandi untuk LDAP DN lookup bind. Amazon MQ mengharuskan penggunaan AWS Secrets Manager untuk menyimpan kata sandi sebagai nilai teks biasa.

Kunci konfigurasi	Deskripsi
`aws.arns.auth_ldap.other_bind.password`	Kata sandi untuk LDAP mengikat lainnya. Amazon MQ mengharuskan penggunaan AWS Secrets Manager untuk menyimpan kata sandi sebagai nilai teks biasa.

Sampel kebijakan IAM

Untuk contoh kebijakan IAM termasuk mengasumsikan dokumen kebijakan peran dan dokumen kebijakan peran, lihat implementasi sampel CDK.

Lihat Menggunakan otentikasi dan otorisasi LDAP langkah-langkah tentang cara mengatur AWS Secrets Manager dan sumber daya Amazon S3.

Validasi akses

Untuk memecahkan masalah skenario di mana nilai ARN tidak dapat diambil, plugin aws mendukung titik akhir API manajemen RabbitMQ yang dapat dipanggil untuk memeriksa apakah Amazon MQ berhasil mengambil peran dan menyelesaikannya. AWS ARNs Ini menghindari kebutuhan untuk memperbarui konfigurasi broker, memperbarui broker dengan revisi konfigurasi baru dan reboot broker untuk menguji perubahan konfigurasi.

> **Note**
>
> Penggunaan API ini memerlukan pengguna administrator RabbitMQ yang sudah ada. Amazon MQ merekomendasikan untuk membuat pialang uji dengan pengguna internal selain metode akses lainnya. Lihat mengaktifkan otentikasi OAuth 2.0 dan sederhana (internal). Pengguna ini kemudian dapat digunakan untuk mengakses API validasi.

> **Note**
>
> Meskipun plugin aws mendukung penerusan peran baru sebagai input ke API validasi, parameter ini tidak didukung oleh Amazon MQ. Peran IAM yang digunakan untuk validasi harus sesuai dengan nilai `aws.arns.assume_role_arn` dalam konfigurasi broker.

Negara karantina broker terkait

Untuk informasi tentang status karantina broker yang terkait dengan masalah dukungan ARN, lihat:

- RABBITMQ_INVALID_ASSUMEROLE
- RABBITMQ_INVALID_ARN_LDAP
- RABBITMQ_INVALID_ARN

Contoh skenario

- Broker `b-f0fc695e-2f9c-486b-845a-988023a3e55b` telah dikonfigurasi untuk menggunakan peran IAM `<role>` untuk mengakses rahasia AWS Secrets Manager `<arn>`
- Jika peran yang diberikan ke Amazon MQ tidak memiliki izin baca pada AWS Secrets Manager rahasia, kesalahan berikut akan ditampilkan di log RabbitMQ:

  ```
  [error] <0.254.0> aws_arn_config: {handle_assume_role,{error,{assume_role_failed,"AWS
   service is unavailable"}}}
  ```

  Selain itu, broker akan memasuki negara `INVALID_ASSUMEROLE` karantina. Untuk informasi lebih lanjut, lihat INVALID_ASSUMEROLE.
- Upaya otentikasi LDAP akan gagal dengan kesalahan berikut:

  ```
  [error] <0.254.0> LDAP bind failed: invalid_credentials
  ```
- Perbaiki peran IAM dengan izin yang tepat
- Panggil titik akhir validasi untuk memverifikasi apakah RabbitMQ sekarang dapat mengakses rahasia:

  ```
  curl -4su 'guest:guest' -XPUT -H 'content-type: application/json' <broker-endpoint>/
  api/aws/arn/validate -d '{"assume_role_arn":"arn:aws:iam::<account-id>:role/<role-
  name>","arns":["arn:aws:secretsmanager:<region>:<account-id>:secret:<secret-name>"]}'
   | jq '.'
  ```

Konfigurasi SSL klien AMQP

Federasi dan sekop menggunakan AMQP untuk komunikasi antara broker hulu dan hilir. Secara default, verifikasi rekan TLS diaktifkan untuk klien AMQP di Amazon MQ untuk RabbitMQ 4. Dengan

pengaturan ini, federasi dan sekop klien AMQP yang berjalan di broker Amazon MQ akan melakukan verifikasi rekan saat membuat koneksi dengan broker hulu.

Klien AMQP yang berjalan di broker Amazon MQ mendukung otoritas sertifikat yang sama dengan Mozilla. Jika Anda tidak menggunakan ACM, gunakan sertifikat yang dikeluarkan oleh CA pada Daftar Sertifikat CA Termasuk Mozilla. Jika broker lokal Anda menggunakan sertifikat dari otoritas sertifikat lain, verifikasi SSL akan gagal. Anda dapat menonaktifkan verifikasi rekan TLS untuk kasus penggunaan ini.

Important

Amazon MQ saat ini tidak mendukung konfigurasi sertifikat klien untuk koneksi klien AMQP. Akibatnya, federasi dan sekop tidak dapat terhubung ke broker berkemampuan MTLS yang memerlukan otentikasi sertifikat klien.

Important

Di Amazon MQ untuk properti SSL RabbitMQ 3 klien AMQP dikonfigurasi dengan default RabbitMQ (verify_none). Amazon MQ untuk RabbitMQ 3 tidak mendukung penggantian default ini.

Note

Dengan `verify_peer` pengaturan default, Anda dapat membuat koneksi federasi dan sekop antara 2 broker MQ Amazon, tetapi ini tidak mendukung pembuatan koneksi antara broker MQ Amazon dan broker swasta atau broker lokal yang menjalankan sertifikat MQ CA non-Amazon. Untuk terhubung dengan broker pribadi atau lokal, Anda perlu menonaktifkan verifikasi rekan di broker Amazon MQ hilir.

Kunci konfigurasi SSL klien AMQP

Konfigurasi	Kunci Konfigurasi	Nilai yang Didukung
Verifikasi rekan SSL klien AMQP	`amqp_client.ssl_options.verify`	`verify_none` , `verify_peer`

Cara mengganti verifikasi rekan SSL klien AMQP

Anda dapat mengganti verifikasi rekan SSL klien AMQP menggunakan API Amazon MQ dan konsol Amazon MQ di broker RabbitMQ 4.

Contoh berikut menunjukkan cara mengganti verifikasi rekan SSL klien AMQP menggunakan: AWS CLI

```
aws mq update-configuration --configuration-id <config-id> --data "$(echo
 "amqp_client.ssl_options.verify=verify_none" | base64 --wrap=0)"
```

Pemanggilan yang berhasil membuat revisi konfigurasi. Anda harus mengaitkan konfigurasi ke broker RabbitMQ Anda dan me-reboot broker untuk menerapkan penggantian. Untuk lebih jelasnya lihat Creating and applying broker configurations

> **Important**
>
> Saat menggunakan`verify_none`, enkripsi SSL masih aktif, tetapi identitas rekan tidak diverifikasi. Gunakan pengaturan ini hanya jika diperlukan dan pastikan bahwa Anda mempercayai jalur jaringan ke broker tujuan.

Amazon MQ untuk Otentikasi dan Otorisasi RabbitMQ

Amazon MQ untuk RabbitMQ mendukung metode otentikasi dan otorisasi berikut:

Otentikasi dan otorisasi sederhana

Dalam metode ini, pengguna broker disimpan secara internal di broker RabbitMQ dan dikelola melalui konsol web atau API manajemen. Izin untuk vhost, pertukaran, antrian, dan topik dikonfigurasi langsung di RabbitMQ. Ini adalah metode default. Untuk informasi selengkapnya, lihat Otentikasi dan otorisasi sederhana.

OAuth 2.0 otentikasi dan otorisasi

Dalam metode ini, pengguna broker dan izin mereka dikelola oleh penyedia identitas OAuth 2.0 eksternal (iDP). Otentikasi pengguna dan izin sumber daya untuk vhost, pertukaran, antrian, dan

topik dipusatkan melalui sistem lingkup penyedia 2.0. OAuth Ini menyederhanakan manajemen pengguna dan memungkinkan integrasi dengan sistem identitas yang ada. Untuk informasi selengkapnya, lihat otentikasi dan otorisasi OAuth 2.0.

Otentikasi dan otorisasi IAM

Dalam metode ini, pengguna broker mengautentikasi menggunakan kredensyal AWS IAM melalui federasi keluar IAM. Kredensyal IAM digunakan untuk mendapatkan token JWT dari AWS Security Token Service (STS), dan token JWT ini berfungsi sebagai token 2.0 untuk otentikasi. OAuth Metode ini memanfaatkan dukungan OAuth 2.0 yang ada di Amazon MQ untuk RabbitMQ, AWS di mana bertindak sebagai penyedia identitas 2.0. OAuth Otentikasi pengguna ditangani oleh AWS IAM, sementara izin sumber daya untuk vhost, pertukaran, antrian, dan topik dikelola melalui kebijakan IAM dan alias ruang lingkup yang dikonfigurasi di RabbitMQ. Untuk informasi selengkapnya, lihat autentikasi dan otorisasi IAM.

Otentikasi dan otorisasi LDAP

Dalam metode ini, pengguna broker dan izin mereka dikelola oleh layanan direktori LDAP eksternal. Otentikasi pengguna dan izin sumber daya dipusatkan melalui server LDAP, memungkinkan pengguna untuk mengakses RabbitMQ menggunakan kredensyal layanan direktori yang ada. Untuk informasi selengkapnya, lihat otentikasi dan otorisasi LDAP.

Otentikasi dan otorisasi HTTP

Dalam metode ini, pengguna broker dan izin mereka dikelola oleh server HTTP eksternal. Otentikasi pengguna dan izin sumber daya dipusatkan melalui server HTTP, memungkinkan pengguna untuk mengakses RabbitMQ menggunakan penyedia Otentikasi dan Otorisasi mereka sendiri. Untuk informasi selengkapnya tentang metode ini, lihat otentikasi dan otorisasi HTTP.

Otentikasi sertifikat SSL

Amazon MQ mendukung TLS bersama (MTLS) untuk broker RabbitMQ. Plugin otentikasi SSL menggunakan sertifikat klien dari koneksi mTLS untuk mengautentikasi pengguna. Dalam metode ini, pengguna broker diautentikasi menggunakan sertifikat klien X.509 alih-alih kredensyal nama pengguna dan kata sandi. Sertifikat klien divalidasi terhadap Otoritas Sertifikat (CA) tepercaya, dan nama pengguna diekstraksi dari bidang dalam sertifikat, seperti Nama Umum (CN) atau Nama Alternatif Subjek (SAN). Metode ini memberikan otentikasi yang kuat tanpa mentransmisikan kredensyal melalui jaringan. Untuk informasi selengkapnya, lihat otentikasi sertifikat SSL.

> **Note**
>
> RabbitMQ mendukung beberapa metode otentikasi dan otorisasi untuk digunakan secara bersamaan. Misalnya, Anda dapat mengaktifkan otentikasi OAuth 2.0 dan sederhana (internal). Untuk informasi lebih lanjut, lihat bagian tutorial OAuth 2.0 tentang mengaktifkan otentikasi OAuth 2.0 dan sederhana (internal) dan dokumentasi kontrol akses RabbitMQ. Amazon MQ merekomendasikan untuk membuat pengguna internal saat menguji konfigurasi otentikasi. Hal ini memungkinkan konfigurasi akses untuk divalidasi menggunakan RabbitMQ management API. Untuk informasi selengkapnya, lihat Validasi akses.

Otentikasi dan otorisasi sederhana

Amazon MQ untuk pengguna broker RabbitMQ

> **Note**
>
> Topik ini menjelaskan pengelolaan pengguna broker dengan otentikasi internal default dan mekanisme otorisasi RabbitMQ. Untuk informasi tentang semua metode autentikasi dan otorisasi yang didukung, lihat Amazon MQ untuk Otentikasi dan Otorisasi RabbitMQ.

Setiap koneksi klien AMQP 0-9-1 memiliki pengguna terkait. Pengguna ini harus diautentikasi. Setiap koneksi klien juga menargetkan host virtual (vhost). Pengguna harus memiliki satu set izin untuk vhost ini. Pengguna mungkin memiliki izin untuk mengonfigurasi, menulis ke, serta membaca dari antrean dan pertukaran di vhost. Anda menentukan kredensyal pengguna dan vhost target saat koneksi dibuat.

Saat pertama kali membuat Amazon MQ untuk broker RabbitMQ, Amazon MQ menggunakan kredensyal masuk yang Anda berikan untuk membuat pengguna RabbitMQ dengan tag tersebut. `administrator` Kemudian Anda dapat menambahkan dan mengelola pengguna melalui API manajemen RabbitMQ atau konsol web RabbitMQ. Anda juga dapat menggunakan konsol web RabbitMQ atau API manajemen untuk mengatur atau memodifikasi izin pengguna dan tanda.

> **Note**
>
> Pengguna RabbitMQ tidak akan disimpan atau ditampilkan melalui API Pengguna Amazon MQ.

> **Important**
>
> Amazon MQ untuk RabbitMQ tidak mendukung nama pengguna “tamu”, dan akan menghapus akun tamu default saat Anda membuat broker baru. Amazon MQ juga akan secara berkala menghapus akun yang dibuat pelanggan yang disebut “tamu”.

Untuk membuat pengguna baru dengan API manajemen RabbitMQ, gunakan titik akhir API berikut dan isi permintaan. Ganti *`username`* dan *`password`* dengan kredensyal masuk baru Anda.

```
PUT /api/users/username HTTP/1.1

    {"password":"password","tags":"administrator"}
```

> **Important**
>
> - Jangan menambahkan informasi identitas pribadi (PII) atau informasi rahasia atau sensitif lainnya dalam nama pengguna broker. Nama pengguna broker dapat diakses oleh AWS layanan lain, termasuk CloudWatch Log. Nama pengguna broker tidak dimaksudkan untuk digunakan untuk data pribadi atau sensitif.
> - Jika Anda kehilangan akses ke semua akun administrator, lihat memulihkan akses broker untuk menggunakan otentikasi IAM untuk pemulihan.

Kunci `tags` adalah hal wajib, dan merupakan daftar tanda yang dipisahkan koma untuk pengguna. Amazon MQ mendukung`administrator`,, `managementmonitoring`, dan tag `policymaker` pengguna.

Anda dapat mengatur izin untuk pengguna individu dengan menggunakan titik akhir API berikut dan isi permintaan. Ganti *`vhost`* dan *`username`* dengan informasi Anda. Untuk vhost default `/`, gunakan `%2F`.

```
PUT /api/permissions/vhost/username HTTP/1.1

    {"configure":".*","write":".*","read":".*"}
```

> **Note**
>
> Kunci `configure`, `read`, dan `write` merupakan hal wajib.

Dengan menggunakan nilai `.*` wildcard, operasi ini akan memberikan pengguna izin membaca, menulis, dan mengonfigurasi untuk semua antrean di vhost yang ditentukan. Untuk informasi selengkapnya tentang mengelola pengguna melalui API manajemen RabbitMQ, lihat HTTP API Manajemen RabbitMQ.

OAuth 2.0 otentikasi dan otorisasi untuk Amazon MQ untuk RabbitMQ

Amazon MQ untuk RabbitMQ mendukung beberapa metode otentikasi dan otorisasi. Untuk informasi tentang semua metode yang didukung, lihat Otentikasi dan otorisasi Amazon MQ untuk broker RabbitMQ.

Dalam otentikasi dan otorisasi OAuth 2.0, pengguna broker dan izin mereka dikelola oleh penyedia identitas OAuth 2.0 eksternal (iDP). Otentikasi pengguna dan izin sumber daya untuk vhost, pertukaran, antrian, dan topik dipusatkan melalui sistem lingkup penyedia 2.0. OAuth Ini menyederhanakan manajemen pengguna dan memungkinkan integrasi dengan sistem identitas yang ada.

> **Pertimbangan penting**
>
> - OAuth Integrasi 2.0 tidak didukung di Amazon MQ untuk broker ActiveMQ.
> - Amazon MQ untuk RabbitMQ tidak mendukung sertifikat server yang dikeluarkan oleh CA pribadi.
> - Plugin RabbitMQ OAuth 2.0 tidak mendukung titik akhir introspeksi token dan token akses buram. Itu juga tidak melakukan pemeriksaan pencabutan token.
> - Anda harus menyertakan izin IAM,`mq:UpdateBrokerAccessConfiguration`, untuk mengaktifkan OAuth 2.0 pada broker yang ada.
> - Amazon MQ secara otomatis membuat pengguna sistem bernama `monitoring-AWS-OWNED-DO-NOT-DELETE` dengan izin pemantauan saja. Pengguna ini menggunakan

sistem otentikasi internal RabbitMQ bahkan pada broker yang OAuth mendukung 2.0 dan dibatasi hanya untuk akses antarmuka loopback.

Untuk informasi tentang cara mengkonfigurasi OAuth 2.0 untuk Amazon MQ Anda untuk broker RabbitMQ, lihat. Menggunakan otentikasi dan otorisasi OAuth 2.0

Di halaman ini

- Konfigurasi OAuth 2.0 yang didukung
- Validasi tambahan untuk otentikasi 2.0 OAuth

Konfigurasi OAuth 2.0 yang didukung

Amazon MQ untuk RabbitMQ mendukung semua variabel yang dapat dikonfigurasi di plugin OAuth RabbitMQ 2.0, dengan pengecualian berikut:

- `auth_oauth2.https.cacertfile`
- `auth_oauth2.oauth_providers.{id/index}.https.cacertfile`
- `management.oauth_client_secret`

  Karena Amazon MQ tidak mendukung kunci ini, kami tidak mendukung UAA sebagai IDP.
- `management.oauth_resource_servers.{id/index}.oauth_client_secret`
- `auth_oauth2.signing_keys.{id/index}`

Validasi tambahan untuk otentikasi 2.0 OAuth

Amazon MQ juga memberlakukan validasi tambahan berikut untuk otentikasi 2.0: OAuth

- Semua URLs harus dimulai dengan`https://`.
- Algoritma tanda tangan yang didukung: `Ed25519Ed25519ph,Ed448,Ed448ph,EdDSA,ES256K,,ES256,ES384,ES512,HS256,HS384,HS512,PS25 RS256RS384`, dan`RS512`.

Otentikasi dan otorisasi IAM untuk Amazon MQ untuk RabbitMQ

Amazon MQ untuk RabbitMQ mendukung beberapa metode otentikasi dan otorisasi. Untuk informasi tentang semua metode yang didukung, lihat Otentikasi dan otorisasi Amazon MQ untuk broker RabbitMQ.

Otentikasi dan otorisasi IAM memungkinkan pengguna broker untuk mengautentikasi menggunakan kredensi IAM melalui federasi AWS keluar IAM. Dalam metode ini, kredensi IAM digunakan untuk mendapatkan token JWT dari AWS Security Token Service (STS). Token JWT ini berfungsi sebagai token OAuth 2.0 untuk otentikasi, memanfaatkan dukungan 2.0 yang ada di Amazon MQ untuk RabbitMQ di mana bertindak sebagai penyedia identitas OAuth 2.0. AWS OAuth AWS IAM menangani otentikasi pengguna, sementara izin sumber daya untuk host virtual, pertukaran, antrian, dan topik dikelola melalui kebijakan IAM dan alias lingkup yang dikonfigurasi di RabbitMQ.

⚠ Pertimbangan penting

- Autentikasi IAM didukung pada RabbitMQ versi 3.13, 4.2 dan di atasnya. Itu tidak didukung di Amazon MQ untuk broker ActiveMQ.
- Autentikasi IAM mengharuskan federasi keluar IAM untuk dikonfigurasi dan tersedia di akun Anda. AWS
- Metode ini dibangun di atas infrastruktur OAuth 2.0 yang ada di Amazon MQ untuk RabbitMQ, AWS dengan berfungsi sebagai penyedia identitas 2.0. OAuth
- Amazon MQ secara otomatis membuat pengguna sistem bernama `monitoring-AWS-OWNED-DO-NOT-DELETE` dengan izin pemantauan saja. Pengguna ini menggunakan sistem otentikasi internal RabbitMQ bahkan pada broker yang mendukung IAM dan dibatasi hanya untuk akses antarmuka loopback.

Di halaman ini

- Cara kerja otentikasi IAM
- Batasan

Cara kerja otentikasi IAM

Otentikasi IAM untuk Amazon MQ untuk RabbitMQ menggunakan federasi keluar IAM untuk mengaktifkan kredensi IAM untuk mengautentikasi dengan broker RabbitMQ. AWS Kredensi IAM

digunakan untuk mendapatkan token JWT dari AWS Security Token Service (STS), dan token JWT ini berfungsi sebagai token OAuth 2.0 untuk otentikasi dengan broker RabbitMQ.

Batasan

Autentikasi IAM untuk Amazon MQ untuk RabbitMQ memiliki batasan sebagai berikut:

- Konfigurasi klaim cakupan — Anda tidak dapat menggunakan klaim cakupan secara langsung karena token JWT dari STS bersarang. Kuncinya adalah`sts.amazonaws.com`, yang mengharuskan penggunaan alias lingkup dalam konfigurasi RabbitMQ untuk memetakan peran IAM ke izin RabbitMQ. Batasan ini juga mencegah penggunaan kebijakan IAM untuk otorisasi sepenuhnya, memerlukan konfigurasi RabbitMQ untuk otorisasi sebagai gantinya.

Untuk informasi tentang cara mengonfigurasi autentikasi dan otorisasi IAM untuk Amazon MQ Anda untuk broker RabbitMQ, lihat. Menggunakan otentikasi dan otorisasi IAM

Otentikasi HTTP dan otorisasi untuk Amazon MQ untuk RabbitMQ

Amazon MQ untuk RabbitMQ mendukung otentikasi dan otorisasi pengguna broker menggunakan server HTTP eksternal. Untuk metode lain yang didukung, lihat Otentikasi dan otorisasi untuk Amazon MQ untuk broker RabbitMQ.

> **Note**
>
> Plugin otentikasi HTTP hanya tersedia untuk Amazon MQ untuk RabbitMQ versi 4 ke atas.

> **Pertimbangan penting**
>
> - Server HTTP harus dapat diakses melalui internet publik. Amazon MQ untuk RabbitMQ dapat dikonfigurasi untuk mengautentikasi ke server HTTP menggunakan TLS timbal balik.
> - Amazon MQ untuk RabbitMQ memberlakukan penggunaan AWS ARNs untuk pengaturan yang memerlukan akses ke sistem file lokal. Lihat dukungan ARN dalam konfigurasi RabbitMQ untuk detail selengkapnya.
> - Anda harus menyertakan izin IAM`mq:UpdateBrokerAccessConfiguration`, untuk mengaktifkan otentikasi HTTP pada broker yang ada.

- Amazon MQ secara otomatis membuat pengguna sistem bernama `monitoring-AWS-OWNED-DO-NOT-DELETE` dengan izin pemantauan saja. Pengguna ini menggunakan sistem otentikasi internal RabbitMQ bahkan pada broker yang mendukung HTTP dan dibatasi hanya untuk akses antarmuka loopback. Amazon MQ mencegah penghapusan pengguna ini dengan menambahkan tag pengguna yang dilindungi.

Untuk informasi tentang cara mengonfigurasi otentikasi HTTP untuk Amazon MQ Anda untuk broker RabbitMQ, lihat. Menggunakan otentikasi dan otorisasi HTTP

Di halaman ini

- Konfigurasi HTTP yang didukung
- Validasi tambahan untuk konfigurasi HTTP di Amazon MQ

Konfigurasi HTTP yang didukung

Amazon MQ untuk RabbitMQ mendukung semua variabel yang dapat dikonfigurasi di plugin otentikasi HTTP RabbitMQ, dengan pengecualian berikut yang memerlukan. AWS ARNs Untuk detail tentang dukungan ARN, lihat dukungan ARN dalam konfigurasi RabbitMQ.

Konfigurasi yang membutuhkan ARNs

`auth_http.ssl_options.cacertfile`

Gunakan `aws.arns.auth_http.ssl_options.cacertfile` sebagai gantinya `auth_http.ssl_options.certfile`

Gunakan `aws.arns.auth_http.ssl_options.certfile` sebagai gantinya `auth_http.ssl_options.keyfile`

Gunakan `aws.arns.auth_http.ssl_options.keyfile` sebagai gantinya

Opsi SSL yang tidak didukung

Opsi konfigurasi SSL berikut juga tidak didukung:

Lihat daftar lengkap

- `auth_http.ssl_options.cert`

- `auth_http.ssl_options.client_renegotiation`
- `auth_http.ssl_options.dh`
- `auth_http.ssl_options.dhfile`
- `auth_http.ssl_options.honor_cipher_order`
- `auth_http.ssl_options.honor_ecc_order`
- `auth_http.ssl_options.key.RSAPrivateKey`
- `auth_http.ssl_options.key.DSAPrivateKey`
- `auth_http.ssl_options.key.PrivateKeyInfo`
- `auth_http.ssl_options.log_alert`
- `auth_http.ssl_options.password`
- `auth_http.ssl_options.psk_identity`
- `auth_http.ssl_options.reuse_sessions`
- `auth_http.ssl_options.secure_renegotiate`
- `auth_http.ssl_options.versions.$version`
- `auth_http.ssl_options.sni`
- `auth_http.ssl_options.crl_check`

Validasi tambahan untuk konfigurasi HTTP di Amazon MQ

Amazon MQ juga memberlakukan validasi tambahan berikut untuk otentikasi dan otorisasi HTTP:

- `auth_http.http_method`harus salah satu `get` atau `post`
- Konfigurasi jalur berikut harus menggunakan HTTPS: URLs
  - `auth_http.user_path`
  - `auth_http.vhost_path`
  - `auth_http.resource_path`
  - `auth_http.topic_path`
- Jika pengaturan apapun memerlukan penggunaan AWS ARN, `aws.arns.assume_role_arn` harus disediakan.

Otentikasi sertifikat SSL untuk Amazon MQ untuk RabbitMQ

Amazon MQ untuk RabbitMQ mendukung otentikasi pengguna broker menggunakan sertifikat klien X.509. Untuk metode lain yang didukung, lihat Otentikasi dan otorisasi untuk Amazon MQ untuk broker RabbitMQ.

> **Note**
>
> Plugin otentikasi sertifikat SSL hanya tersedia untuk Amazon MQ untuk RabbitMQ versi 4 ke atas.

> **Pertimbangan penting**
>
> - Sertifikat klien harus ditandatangani oleh Otoritas Sertifikat (CA) tepercaya. Amazon MQ untuk RabbitMQ memvalidasi rantai sertifikat selama otentikasi.
> - Amazon MQ untuk RabbitMQ memberlakukan penggunaan AWS ARNs untuk pengaturan terkait sertifikat seperti sertifikat CA dan untuk pengaturan yang memerlukan akses ke sistem file lokal. Lihat dukungan ARN dalam konfigurasi RabbitMQ untuk detail selengkapnya.
> - Amazon MQ secara otomatis membuat pengguna sistem bernama `monitoring-AWS-OWNED-DO-NOT-DELETE` dengan izin pemantauan saja. Pengguna ini menggunakan sistem otentikasi internal RabbitMQ bahkan pada broker yang mendukung sertifikat SSL dan dibatasi hanya untuk akses antarmuka loopback. Amazon MQ mencegah penghapusan pengguna ini dengan menambahkan tag pengguna yang dilindungi.

Untuk informasi tentang cara mengonfigurasi otentikasi sertifikat SSL untuk Amazon MQ Anda untuk broker RabbitMQ, lihat. Menggunakan otentikasi sertifikat SSL

Di halaman ini

- Konfigurasi SSL yang didukung
- Validasi tambahan untuk konfigurasi SSL di Amazon MQ

Konfigurasi SSL yang didukung

Amazon MQ untuk RabbitMQ mendukung SSL/TLS konfigurasi untuk koneksi klien. Untuk detail tentang dukungan ARN, lihat dukungan ARN dalam konfigurasi RabbitMQ.

Konfigurasi yang membutuhkan ARNs

`ssl_options.cacertfile`

Gunakan `aws.arns.ssl_options.cacertfile` sebagai gantinya

Konfigurasi login sertifikat SSL

Konfigurasi berikut mengontrol cara nama pengguna diekstraksi dari sertifikat klien:

`ssl_cert_login_from`

Menentukan bidang sertifikat yang akan digunakan untuk ekstraksi nama pengguna. Nilai yang didukung:

- `distinguished_name`- Gunakan Nama Distinguished lengkap
- `common_name`- Gunakan bidang Nama Umum (CN)
- `subject_alternative_name`atau `subject_alt_name` - Gunakan Nama Alternatif Subjek

`ssl_cert_login_san_type`

Saat menggunakan Nama Alternatif Subjek, menentukan jenis SAN. Nilai yang didukung:`dns`,`ip`,`email`,`uri`, `other_name`

`ssl_cert_login_san_index`

Saat menggunakan Nama Alternatif Subjek, menentukan indeks entri SAN yang akan digunakan (berbasis nol). Harus berupa bilangan bulat non-negatif.

Opsi SSL untuk koneksi klien

Opsi SSL berikut berlaku untuk koneksi klien:

`ssl_options.verify`

Mode verifikasi rekan. Nilai yang didukung:`verify_none`, `verify_peer`

`ssl_options.fail_if_no_peer_cert`

Apakah akan menolak koneksi jika klien tidak memberikan sertifikat. Nilai Boolean.

`ssl_options.depth`

Kedalaman rantai sertifikat maksimum untuk verifikasi.

`ssl_options.hostname_verification`

Mode verifikasi nama host. Nilai yang didukung:`wildcard`, `none`

Opsi SSL yang tidak didukung

Opsi konfigurasi SSL berikut tidak didukung:

Lihat daftar lengkap

- `ssl_options.cert`
- `ssl_options.client_renegotiation`
- `ssl_options.dh`
- `ssl_options.dhfile`
- `ssl_options.honor_cipher_order`
- `ssl_options.honor_ecc_order`
- `ssl_options.key.RSAPrivateKey`
- `ssl_options.key.DSAPrivateKey`
- `ssl_options.key.PrivateKeyInfo`
- `ssl_options.log_alert`
- `ssl_options.password`
- `ssl_options.psk_identity`
- `ssl_options.reuse_sessions`
- `ssl_options.secure_renegotiate`
- `ssl_options.versions.$version`
- `ssl_options.sni`
- `ssl_options.crl_check`

Validasi tambahan untuk konfigurasi SSL di Amazon MQ

Amazon MQ juga memberlakukan validasi tambahan berikut untuk otentikasi sertifikat SSL:

- Jika pengaturan apapun memerlukan penggunaan AWS ARN, `aws.arns.assume_role_arn` harus disediakan.

Otentikasi dan otorisasi LDAP untuk Amazon MQ untuk RabbitMQ

Amazon MQ untuk RabbitMQ mendukung otentikasi dan otorisasi pengguna broker menggunakan server LDAP eksternal. Untuk metode lain yang didukung, lihat Otentikasi dan otorisasi untuk Amazon MQ untuk broker RabbitMQ.

⚠ Pertimbangan penting

- Server LDAP harus dapat diakses melalui internet publik. Amazon MQ untuk RabbitMQ dapat dikonfigurasi untuk mengautentikasi ke server LDAP menggunakan TLS timbal balik.
- Amazon MQ untuk RabbitMQ memberlakukan penggunaan AWS ARNs untuk pengaturan LDAP sensitif seperti kata sandi dan untuk pengaturan yang memerlukan akses ke sistem file lokal. Lihat dukungan ARN dalam konfigurasi RabbitMQ untuk detail selengkapnya.
- Anda harus menyertakan izin IAM,`mq:UpdateBrokerAccessConfiguration`, untuk mengaktifkan LDAP pada broker yang ada.
- Amazon MQ secara otomatis membuat pengguna sistem bernama `monitoring-AWS-OWNED-DO-NOT-DELETE` dengan izin pemantauan saja. Pengguna ini menggunakan sistem otentikasi internal RabbitMQ bahkan pada broker yang mendukung LDAP dan dibatasi hanya untuk akses antarmuka loopback. Amazon MQ mencegah penghapusan pengguna ini dengan menambahkan tag pengguna yang dilindungi.

Untuk informasi tentang cara mengkonfigurasi LDAP untuk Amazon MQ Anda untuk broker RabbitMQ, lihat. Menggunakan otentikasi dan otorisasi LDAP

Di halaman ini

- Konfigurasi LDAP yang didukung
- Validasi tambahan untuk konfigurasi LDAP di Amazon MQ

Konfigurasi LDAP yang didukung

Amazon MQ untuk RabbitMQ mendukung semua variabel yang dapat dikonfigurasi di plugin RabbitMQ LDAP, dengan pengecualian berikut yang memerlukan. AWS ARNs Untuk detail tentang dukungan ARN, lihat dukungan ARN dalam konfigurasi RabbitMQ.

Konfigurasi yang membutuhkan ARNs

`auth_ldap.dn_lookup_bind.password`

Gunakan `aws.arns.auth_ldap.dn_lookup_bind.password` sebagai gantinya

`auth_ldap.other_bind.password`

Gunakan `aws.arns.auth_ldap.other_bind.password` sebagai gantinya

`auth_ldap.ssl_options.cacertfile`

Gunakan `aws.arns.auth_ldap.ssl_options.cacertfile` sebagai gantinya

`auth_ldap.ssl_options.certfile`

Gunakan `aws.arns.auth_ldap.ssl_options.certfile` sebagai gantinya

`auth_ldap.ssl_options.keyfile`

Gunakan `aws.arns.auth_ldap.ssl_options.keyfile` sebagai gantinya

Opsi SSL yang tidak didukung

Opsi konfigurasi SSL berikut juga tidak didukung:

Lihat daftar lengkap

- `auth_ldap.ssl_options.cert`
- `auth_ldap.ssl_options.client_renegotiation`
- `auth_ldap.ssl_options.dh`
- `auth_ldap.ssl_options.dhfile`
- `auth_ldap.ssl_options.honor_cipher_order`
- `auth_ldap.ssl_options.honor_ecc_order`
- `auth_ldap.ssl_options.key.RSAPrivateKey`

- `auth_ldap.ssl_options.key.DSAPrivateKey`
- `auth_ldap.ssl_options.key.PrivateKeyInfo`
- `auth_ldap.ssl_options.log_alert`
- `auth_ldap.ssl_options.password`
- `auth_ldap.ssl_options.psk_identity`
- `auth_ldap.ssl_options.reuse_sessions`
- `auth_ldap.ssl_options.secure_renegotiate`
- `auth_ldap.ssl_options.versions.$version`
- `auth_ldap.ssl_options.sni`

Validasi tambahan untuk konfigurasi LDAP di Amazon MQ

Amazon MQ juga memberlakukan validasi tambahan berikut untuk otentikasi dan otorisasi LDAP:

- `auth_ldap.log`tidak dapat diatur ke `network_unsafe`
- Server LDAP harus menggunakan LDAPS. Entah `auth_ldap.use_ssl` atau `auth_ldap.use_starttls` harus diaktifkan secara eksplisit
- Jika pengaturan apapun memerlukan penggunaan AWS ARN, `aws.arns.assume_role_arn` harus disediakan.
- `auth_ldap.servers`harus berupa alamat IP yang valid atau FQDN yang valid
- Kunci berikut harus berupa Nama Distinguished LDAP yang valid:
  - `auth_ldap.dn_lookup_base`
  - `auth_ldap.dn_lookup_bind.user_dn`
  - `auth_ldap.other_bind.user_dn`
  - `auth_ldap.group_lookup_base`

Plugin

Amazon MQ untuk RabbitMQ juga mendukung plugin berikut.

- Plugin manajemen RabbitMQ
- Plugin sekop

- Plugin Federasi
- Plugin pertukaran Hash yang konsisten
- OAuth Plugin 2
- Plugin LDAP
- Plugin HTTP
- Plugin sertifikat SSL
- plugin aws
- Plugin Pertukaran Topik JMS

Plugin manajemen RabbitMQ

Amazon MQ untuk RabbitMQ mendukung plugin manajemen RabbitMQ, yang menyediakan API manajemen berbasis HTTP bersama dengan UI berbasis browser untuk konsol web RabbitMQ. Anda dapat menggunakan konsol web dan API manajemen untuk membuat serta mengelola pengguna dan kebijakan broker.

Plugin shovel

Amazon MQ untuk RabbitMQ mendukung plugin sekop RabbitMQ, yang memungkinkan Anda memindahkan pesan dari antrian dan pertukaran pada satu broker ke broker lainnya. Anda dapat menggunakan shovel untuk menghubungkan broker dengan penggabungan longgar dan mendistribusikan pesan dari simpul dengan beban pesan yang lebih berat.

> **Important**
>
> Anda tidak dapat mengonfigurasi sekop di antara antrian atau pertukaran jika tujuan sekop adalah broker pribadi.
> Amazon MQ tidak mendukung penggunaan shovel statis.

Hanya sekop dinamis yang didukung. Sekop dinamis dikonfigurasi menggunakan parameter runtime dan dapat dimulai dan dihentikan kapan saja secara terprogram oleh koneksi klien. Misalnya, menggunakan API manajemen RabbitMQ, Anda dapat membuat permintaan PUT ke titik akhir API berikut untuk mengonfigurasi sekop dinamis. Dalam contoh, {vhost} dapat diganti dengan nama vhost broker, dan {name} diganti dengan nama sekop dinamis baru.

```
/api/parameters/shovel/{vhost}/{name}
```

Dalam isi permintaan, Anda harus menentukan antrean atau pertukaran, tidak keduanya. Contoh di bawah ini mengonfigurasi sekop dinamis antara antrian lokal yang ditentukan dalam src-queue dan antrian jarak jauh yang ditentukan dalam dest-queue. Demikian pula, Anda dapat menggunakan parameter src-exchange dan dest-exchange untuk mengkonfigurasi sekop antara dua bursa.

```
{
"value": {
"src-protocol": "amqp091",
"src-uri": "amqp://localhost",
"src-queue": "source-queue-name",
"dest-protocol": "amqp091",
"dest-uri": "amqps://b-c8352341-ec91-4a78-ad9c-a43f23d325bb.mq.us-
west2.amazonaws.com:5671",
"dest-queue": "destination-queue-name"
}
}
```

Plugin federasi

Amazon MQ mendukung pertukaran dan antrian federasi menggunakan plugin federasi RabbitMQ. Dengan federasi, Anda dapat mereplikasi alur pesan antara antrean, pertukaran, dan konsumen pada broker terpisah. Antrian dan pertukaran federasi menggunakan point-to-point tautan untuk terhubung ke rekan-rekan di broker lain. Sedangkan pertukaran federasi, secara default, merutekan pesan satu kali, antrean federasi dapat memindahkan pesan beberapa kali sesuai kebutuhan konsumen.

Anda dapat menggunakan federasi untuk memungkinkan broker hilir mengonsumsi pesan dari pertukaran atau antrean di hulu. Anda dapat mengaktifkan federasi di broker hilir menggunakan konsol web RabbitMQ atau API manajemen.

> **Important**
>
> Anda tidak dapat mengonfigurasi federasi jika antrian atau pertukaran hulu ada di broker pribadi. Anda hanya dapat mengkonfigurasi federasi antara antrian atau pertukaran di pialang publik, atau antara antrian hulu atau pertukaran di broker publik, dan antrian hilir atau pertukaran di broker swasta.

Misalnya, menggunakan API manajemen, Anda dapat mengonfigurasi federasi dengan melakukan hal berikut.

- Mengonfigurasi satu atau lebih hulu yang menentukan koneksi federasi ke simpul lain. Anda dapat menentukan koneksi federasi menggunakan konsol web RabbitMQ atau API manajemen. Menggunakan API manajemen, Anda dapat membuat permintaan POST ke/api/parameters/federation-upstream/%2f/myupstreamdengan badan permintaan berikut.

  ```
  {"value":{"uri":"amqp://server-name","expires":3600000}}
  ```

- Konfigurasikan kebijakan untuk mengaktifkan antrean atau pertukaran agar menjadi federasi. Anda dapat mengonfigurasi kebijakan menggunakan konsol web RabbitMQ atau API manajemen. Menggunakan API manajemen, Anda dapat membuat permintaan POST ke/api/policies/%2f/federate-me dengan badan permintaan berikut.

  ```
  {"pattern":"^amq\.", "definition":{"federation-upstream-set":"all"}, "apply-to":"exchanges"}
  ```

  > **Note**
  >
  > Badan permintaan mengasumsikan pertukaran di server diberi nama dimulai dengan amq. Menggunakan ekspresi reguler ^amq\. akan memastikan bahwa federasi diaktifkan untuk semua bursa yang namanya dimulai dengan “amq.” Pertukaran di server RabbitMQ Anda dapat diberikan nama yang berbeda.

Plugin pertukaran Hash yang konsisten

Amazon MQ untuk RabbitMQ mendukung plugin RabbitMQ Consistent Hash Exchange Type. Pertukaran Hash yang konsisten merutekan pesan ke antrian berdasarkan nilai hash yang dihitung dari kunci perutean pesan. Mengingat kunci routing yang cukup merata, pertukaran Hash Konsisten dapat mendistribusikan pesan antar antrian secara merata.

Untuk antrian yang terikat pada pertukaran Hash Konsisten, kunci pengikatan adalah kunci number-as-a-string yang menentukan bobot pengikatan setiap antrian. Antrian dengan bobot pengikatan yang lebih tinggi akan menerima distribusi pesan yang lebih tinggi secara proporsional dari pertukaran Hash Konsisten yang terikat. Dalam topologi pertukaran Hash Konsisten, penerbit dapat dengan

mudah mempublikasikan pesan ke bursa, tetapi konsumen harus dikonfigurasi secara eksplisit untuk mengkonsumsi pesan dari antrian tertentu.

OAuth Plugin 2.0

Amazon MQ untuk RabbitMQ mendukung plugin backend otentikasi 2. OAuth Plugin ini diaktifkan secara kondisional berdasarkan konfigurasi broker Anda. Ketika diaktifkan, plugin ini menyediakan otentikasi dan otorisasi OAuth 2.0 dengan integrasi ke penyedia identitas OAuth 2.0 eksternal untuk manajemen pengguna terpusat dan kontrol akses. Untuk informasi selengkapnya tentang otentikasi OAuth 2.0, lihatOAuth 2.0 otentikasi dan otorisasi.

Plugin LDAP

Amazon MQ untuk RabbitMQ mendukung plugin backend otentikasi LDAP. Plugin ini diaktifkan secara kondisional berdasarkan konfigurasi broker Anda. Ketika diaktifkan, plugin ini menyediakan otentikasi dan otorisasi LDAP dengan integrasi ke layanan direktori LDAP eksternal untuk otentikasi dan otorisasi pengguna terpusat. Untuk informasi selengkapnya tentang otentikasi LDAP, lihat. Otentikasi dan otorisasi LDAP

Plugin HTTP

Amazon MQ untuk RabbitMQ mendukung plugin backend otentikasi HTTP. Plugin ini diaktifkan secara kondisional berdasarkan konfigurasi broker Anda. Ketika diaktifkan, plugin ini menyediakan otentikasi HTTP dan otorisasi dengan integrasi ke server HTTP eksternal untuk otentikasi dan otorisasi pengguna terpusat. Untuk informasi selengkapnya tentang otentikasi HTTP, lihatOtentikasi dan otorisasi HTTP.

> **Note**
>
> Plugin otentikasi HTTP hanya tersedia untuk Amazon MQ untuk RabbitMQ versi 4 ke atas.

Plugin sertifikat SSL

Amazon MQ mendukung TLS bersama (MTLS) untuk broker RabbitMQ. Plugin otentikasi SSL menggunakan sertifikat klien dari koneksi mTLS untuk mengautentikasi pengguna. Plugin ini diaktifkan secara kondisional berdasarkan konfigurasi broker Anda. Saat diaktifkan, ia menyediakan otentikasi berbasis sertifikat menggunakan sertifikat klien X.509 untuk otentikasi yang kuat tanpa

mengirimkan kredensyal melalui jaringan. Untuk informasi selengkapnya tentang otentikasi sertifikat SSL, lihat. Otentikasi sertifikat SSL

> **Note**
>
> Plugin otentikasi sertifikat SSL hanya tersedia untuk Amazon MQ untuk RabbitMQ versi 4 ke atas.

plugin aws

Plugin aws diaktifkan secara kondisional oleh Amazon MQ untuk RabbitMQ berdasarkan konfigurasi broker Anda. Plugin komunitas ini, dikembangkan dan dikelola oleh Amazon MQ, menyediakan pengambilan kredensi dan sertifikat yang aman dari AWS layanan yang digunakan AWS ARNs dalam pengaturan konfigurasi RabbitMQ. Untuk informasi lebih lanjut tentang dukungan ARN, lihat. ARN support in RabbitMQ configuration

Plugin Pertukaran Topik JMS

Plugin Pertukaran Topik JMS selalu diaktifkan oleh Amazon MQ untuk RabbitMQ. Ia bekerja dengan klien RabbitMQ JMS untuk memungkinkan aplikasi JMS baru dan yang sudah ada terhubung ke Amazon MQ untuk RabbitMQ.

> **Note**
>
> Plugin JMS Topic Exchange hanya tersedia untuk Amazon MQ untuk RabbitMQ versi 4 ke atas. Ini diaktifkan secara default tetapi hanya mengaktifkan ketika klien RabbitMQ JMS digunakan untuk menjalankan beban kerja JMS.

Protokol yang didukung

Anda dapat mengakses broker RabbitMQ Anda dengan menggunakan bahasa pemrograman apa pun yang didukung RabbitMQ dan dengan mengaktifkan TLS untuk salah satu spesifikasi protokol berikut:

- AMQP (0-9-1)
- AMQP 1.0

- JMS 1.1
- JMS 2.0
- JMS 3.1

Amazon MQ untuk dukungan RabbitMQ JMS

Anda sekarang dapat menjalankan beban kerja JMS 1.1, 2.0, dan 3.1 di Amazon MQ untuk RabbitMQ 4 dengan klien RabbitMQ JMS.

Klien RabbitMQ JMS

Klien RabbitMQ JMS adalah pustaka klien JMS open-source yang Anda perlukan untuk menghubungkan aplikasi JMS Anda ke broker Amazon MQ RabbitMQ. Untuk informasi lebih lanjut, silakan kunjungi GitHub repositori resmi.

Didukung JMS 1.1, 2.0 dan 3.1 APIs

Dari Amazon MQ untuk RabbitMQ 4 dan seterusnya, plugin selalu diaktifkan. `jms-topic-exchange` Oleh karena itu, Anda dapat menggunakan Amazon MQ untuk klien RabbitMQ 4 dan RabbitMQ JMS untuk beban kerja JMS Anda. Semua JMS APIs didefinisikan dalam JMS 1.1 didukung kecuali:

- Sesi APIs server tidak didukung.
- Transaksi XA APIs tidak didukung.
- Pemilih JMS untuk tujuan Antrian JMS tidak didukung.
- Atribut `NoLocal` langganan JMS tidak didukung.

Semua yang baru ditambahkan APIs di JMS 2.0 dan JMS 3.1 didukung kecuali:

- `JMSProducer.setDeliveryDelay`API tidak didukung.

Untuk mempelajari lebih lanjut tentang menghubungkan aplikasi JMS Anda ke Amazon MQ untuk broker RabbitMQ, silakan lihat tutorial tentang Menghubungkan aplikasi JMS Anda ke Amazon MQ untuk broker RabbitMQ

Autentikasi dan Otorisasi

Semua mekanisme otentikasi dan otorisasi yang tercantum dalam bagian ini didukung. Kredensi yang digunakan untuk menghubungkan ke broker menggunakan klien JMS sama seperti jika Anda terhubung ke broker RabbitMQ menggunakan klien AMQP Java.

Interoperabilitas dengan antrian AMQP di RabbitMQ

Anda dapat menggunakan klien RabbitMQ JMS untuk mengirim pesan JMS ke pertukaran AMQP dan menggunakan pesan dari antrian AMQP (fitur ini tidak mendukung topik JMS). Ini memungkinkan Anda untuk menginteroperasikan atau memigrasikan beban kerja JMS tertentu ke beban kerja AMQP. Untuk informasi lebih lanjut, silakan kunjungi dokumentasi klien resmi.

Menerapkan kebijakan ke Amazon MQ untuk RabbitMQ

Anda dapat menerapkan kebijakan dan batasan khusus dengan nilai default yang direkomendasikan Amazon MQ. Jika Anda telah menghapus kebijakan default dan batas yang direkomendasikan, lalu ingin membuat ulang kebijkan dan batas, atau Anda telah membuat vhost tambahan dan ingin menerapkan kebijakan default dan batas ke vhosts baru, Anda dapat menggunakan langkah-langkah berikut.

Important

Di Amazon MQ untuk mesin RabbitMQ versi 3.13 dan di bawahnya, kebijakan operator default saat ini adalah:

```
vhost name pattern apply-to definition priority/
 default_operator_policy_AWS_managed .* classic_queues {"ha-mode":"all","ha-
sync-mode":"automatic","queue-version":2} 0
```

Pada versi 4.0 dan di atasnya, kebijakan operator default telah berubah menjadi:

```
vhost name pattern apply-to definition priority/
 default_operator_policy_AWS_managed .* classic_queues {"queue-version":2} 0
```

Perubahan ini diperlukan karena pencerminan antrian klasik dan pengaturan kebijakan HA tidak didukung di RabbitMQ 4.
Anda tidak dapat membuat kebijakan yang berlaku untuk antrian cermin klasik dan antrian kuorum. Jika Anda ingin kebijakan Anda hanya berlaku untuk antrian kuorum, Anda harus

menyetel ke. `--apply-to quorum_queues` Jika Anda menggunakan antrian cermin klasik dan antrian kuorum, Anda harus membuat kebijakan terpisah dengan `--apply-to:classic_queues` serta kebijakan antrian kuorum.

Important

Untuk melakukan langkah-langkah berikut, Anda harus memiliki pengguna broker Amazon MQ for RabbitMQ dengan izin administrator. Anda dapat menggunakan pengguna administrator yang dibuat ketika pertama kali membuat broker, atau pengguna lain yang mungkin telah Anda buat sesudahnya. Tabel berikut menyediakan tanda pengguna administrator yang diperlukan dan izin sebagai pola ekspresi reguler (regexp).

Tanda	Baca regexp	Konfigurasikan regexp	Tulis regexp
`administrator`	`.*`	`.*`	`.*`

Untuk informasi selengkapnya tentang cara membuat pengguna RabbitMQ serta mengelola tanda dan izin pengguna, lihat Amazon MQ untuk pengguna broker RabbitMQ.

Untuk menerapkan kebijakan default dan batas host virtual menggunakan konsol web RabbitMQ

1. Masuk ke konsol Amazon MQ.
2. Di panel navigasi kiri, pilih Broker.
3. Dari daftar broker, pilih nama broker yang ingin Anda terapkan kebijakan baru.
4. Di halaman detail broker, pada bagian Koneksi, pilih URL konsol web RabbitMQ. Konsol web RabbitMQ terbuka di tab browser atau jendela baru.
5. Login ke konsol web RabbitMQ dengan nama pengguna dan kata sandi administrator broker Anda.
6. Di konsol web RabbitMQ, di bagian atas halaman, pilih Admin.
7. Di halaman Admin, di panel navigasi kanan, pilih Kebijakan.
8. Di halaman Kebijakan, Anda dapat melihat daftar Kebijakan pengguna broker saat ini. Di bawah Kebijakan pengguna, perluas Tambahkan / perbarui kebijakan.

9. Untuk membuat kebijakan broker baru, di bawah Tambahkan / perbarui kebijakan, lakukan hal berikut:

   a. Untuk Host virtual, pilih nama vhost yang ingin dilampirkan kebijakan dari daftar dropdown. Untuk memilih vhost default, pilih /.

   > **Note**
   >
   > Jika Anda belum membuat vhost tambahan, opsi Host virtual tidak ditampilkan pada konsol RabbitMQ, dan kebijakan diterapkan hanya untuk vhost default.

   b. Untuk Nama, masukkan nama kebijakan Anda, misalnya **`policy-defaults`**.

   c. Untuk Pola, masukkan pola regexp **`.*`** sehingga kebijakan cocok dengan semua antrean pada broker.

   d. Untuk Terapkan ke, pilih Pertukaran dan antrean dari daftar dropdown.

   e. Untuk Prioritas, masukkan bilangan bulat yang lebih besar dari semua kebijakan lain yang diterapkan ke vhost. Anda dapat menerapkan satu set definisi kebijakan ke antrean dan pertukaran RabbitMQ pada waktu tertentu. RabbitMQ memilih kebijakan yang cocok dengan nilai prioritas tertinggi. Untuk informasi selengkapnya tentang prioritas kebijakan dan cara menggabungkan kebijakan, lihat Kebijakan dalam Dokumentasi Server RabbitMQ.

   f. Untuk Definisi, tambahkan pasangan nilai kunci berikut:

   - **`queue-mode=lazy`**. Pilih String dari daftar dropdown.
   - **`overflow=reject-publish`**. Pilih String dari daftar dropdown.

     > **Note**
     >
     > Tidak berlaku untuk broker instans tunggal.

   - **`max-length=`*`number-of-messages`***. Ganti *`number-of-messages`* dengan nilai yang direkomendasikan Amazon MQ sesuai dengan ukuran instans dan mode penerapan broker, misalnya, **`8000000`** untuk klaster. `mq.m7g.large` Pilih Nomor dari daftar dropdown.

     > **Note**
     >
     > Tidak berlaku untuk broker instans tunggal.

g. Pilih Buat / perbarui kebijakan.

10. Konfirmasi bahwa kebijakan baru muncul dalam daftar Kebijakan pengguna.

> **Note**
>
> Untuk broker klaster, Amazon MQ secara otomatis menerapkan definisi kebijakan `ha-mode: all` dan `ha-sync-mode: automatic`.

11. Dari panel navigasi kanan, pilih Batas.
12. Di halaman Batas, Anda dapat melihat daftar Batas host virtual broker saat ini. Di bawah Batas host virtual, perluas Atur / perbarui batas host virtual.
13. Untuk membuat batas vhost baru, di bawah Atur / perbarui batas host virtual, lakukan hal berikut:

    a. Untuk Host virtual, pilih nama vhost yang ingin dilampirkan kebijakan dari daftar dropdown. Untuk memilih vhost default, pilih /.

    b. Untuk Batas, pilih max-connections dari opsi dropdown.

    c. Untuk Nilai, masukkan nilai yang direkomendasikan Amazon MQ sesuai dengan ukuran instans broker dan mode deployment, misalnya, **`15000`** untuk klaster `mq.m5.large`.

    d. Pilih Atur / perbarui batas.

    e. Ulangi langkah di atas, dan untuk Batas, pilih max-queues dari opsi dropdown.

14. Konfirmasikan bahwa batas baru muncul dalam daftar Batas host virtual.

Untuk menerapkan kebijakan default dan batas host virtual menggunakan API manajemen RabbitMQ

1. Masuk ke konsol Amazon MQ.
2. Di panel navigasi kiri, pilih Broker.
3. Dari daftar broker, pilih nama broker yang ingin Anda terapkan kebijakan baru.
4. Di halaman broker, pada bagian Koneksi, catat URL konsol web RabbitMQ. Ini adalah titik akhir broker yang Anda gunakan dalam permintaan HTTP.
5. Buka terminal atau jendela baris perintah baru pilihan Anda.
6. Untuk membuat kebijakan broker baru, masukkan perintah `curl` baru. Perintah ini mengasumsikan antrean pada vhost `/` default, yang dienkode sebagai `%2F`. Untuk menerapkan kebijakan ke vhost lain, ganti `%2F` dengan nama vhost.

> **Note**
>
> Ganti *username* dan *password* dengan kredenal masuk administrator Anda. Ganti *number-of-messages* dengan nilai yang direkomendasikan Amazon MQ sesuai dengan ukuran instans dan mode penerapan broker. Ganti *policy-name* dengan nama untuk kebijakan Anda. Ganti *broker-endpoint* dengan URL yang Anda catat sebelumnya.

```
curl -i -u username:password -H "content-type:application/json" -XPUT \
-d '{"pattern":".*", "priority":1, "definition":{"queue-mode":lazy,
 "overflow":"reject-publish", "max-length":"number-of-messages"}}' \
broker-endpoint/api/policies/%2F/policy-name
```

7. Untuk mengonfirmasi bahwa kebijakan baru ditambahkan ke kebijakan pengguna broker, masukkan perintah `curl` berikut untuk daftar seluruh kebijakan broker.

```
curl -i -u username:password broker-endpoint/api/policies
```

8. Untuk membuat batas host virtual `max-connections` yang baru, masukkan perintah `curl` berikut. Perintah ini mengasumsikan antrean pada vhost `/` default, yang dienkode sebagai `%2F`. Untuk menerapkan kebijakan ke vhost lain, ganti `%2F` dengan nama vhost.

> **Note**
>
> Ganti *username* dan *password* dengan kredenal masuk administrator Anda. Ganti *max-connections* dengan nilai yang direkomendasikan Amazon MQ sesuai dengan ukuran instans dan mode penerapan broker. Mengganti titik akhir broker dengan URL yang Anda catat sebelumnya.

```
curl -i -u username:password -H "content-type:application/json" -XPUT \
-d '{"value":"number-of-connections"}' \
broker-endpoint/api/vhost-limits/%2F/max-connections
```

9. Untuk membuat batas host virtual `max-queues`, ulangi langkah sebelumnya, tetapi modifikasi perintah curl seperti yang ditampilkan berikut ini.

```
curl -i -u username:password -H "content-type:application/json" -XPUT \
-d '{"value":"number-of-queues"}' \
broker-endpoint/api/vhost-limits/%2F/max-queues
```

10. Untuk mengonfirmasi bahwa batas baru ditambahkan ke batas host virtual broker Anda, masukkan perintah `curl` berikut untuk memuat daftar semua batas host virtual broker.

```
curl -i -u username:password broker-endpoint/api/vhost-limits
```

Antrian kuorum untuk RabbitMQ di Amazon MQ

Antrian kuorum adalah jenis antrian yang direplikasi yang terdiri dari pemimpin (replika utama) dan pengikut (replika lainnya). Jika pemimpin menjadi tidak tersedia, antrian kuorum menggunakan algoritma konsensus Raft untuk memilih node pemimpin baru dengan mayoritas suara, dan pemimpin sebelumnya diturunkan ke node pengikut di cluster yang sama. Pengikut yang tersisa terus mereplikasi seperti sebelumnya. Karena setiap node berada di zona ketersediaan yang berbeda, jika satu node sementara tidak tersedia, pengiriman pesan berlanjut dengan replika pemimpin yang baru dipilih di zona ketersediaan lain.

Antrian kuorum berguna untuk menangani pesan racun, yang terjadi ketika pesan gagal dan diminta ulang beberapa kali.

Anda tidak boleh menggunakan antrian kuorum jika Anda:

- gunakan antrian sementara
- memiliki backlog antrian panjang
- memprioritaskan latensi rendah

Untuk mendeklarasikan antrian kuorum, atur header ke. `x-queue-type quorum`

Topik

- Bermigrasi dari antrian klasik ke antrian kuorum di Amazon MQ untuk RabbitMQ
- Konfigurasi kebijakan untuk antrian kuorum untuk Amazon MQ untuk RabbitMQ

- Praktik terbaik untuk antrian kuorum untuk Amazon MQ untuk RabbitMQ

Bermigrasi dari antrian klasik ke antrian kuorum di Amazon MQ untuk RabbitMQ

Anda dapat memigrasikan antrian cermin klasik Anda ke antrian kuorum di broker Amazon MQ pada versi 3.13 atau lebih tinggi dengan membuat host virtual baru di cluster yang sama, atau dengan bermigrasi di tempat.

Opsi 1: Bermigrasi dari antrian cermin klasik ke antrian kuorum dengan host virtual baru

Anda dapat memigrasikan antrian cermin klasik Anda ke antrian kuorum di broker Amazon MQ pada versi 3.13 atau lebih tinggi dengan membuat host virtual baru di cluster yang sama.

1. Di cluster Anda yang ada, buat host virtual baru (vhost) dengan tipe antrian default sebagai kuorum.
2. Buat Plugin federasi dari vhost baru dengan URI menunjuk ke vhost lama menggunakan antrian cermin klasik.
3. Menggunakan`rabbitmqadmin`, ekspor definisi dari vhost lama ke file baru. Anda harus membuat perubahan pada file skema sehingga kompatibel dengan antrian kuorum. Untuk daftar lengkap perubahan yang perlu Anda buat pada file, lihat Memindahkan definisi dalam dokumentasi antrian kuorum RabbitMQ. Setelah menerapkan perubahan yang diperlukan pada file, impor ulang definisi ke vhost baru.
4. Buat kebijakan baru di vhost baru. Untuk rekomendasi tentang konfigurasi kebijakan Amazon MQ untuk antrian kuorum, lihat. Konfigurasi kebijakan untuk antrian kuorum untuk Amazon MQ untuk RabbitMQ Kemudian, mulai Federasi yang Anda buat sebelumnya dari vhost lama ke vhost baru.
5. Arahkan konsumen dan produsen ke vhost baru.
6. Konfigurasikan steker Sekop untuk memindahkan pesan yang tersisa. Setelah antrian kosong, hapus Shovel.

Bermigrasi dari antrian cermin klasik ke antrian kuorum di tempat

Anda dapat memigrasikan antrian cermin klasik Anda ke antrian kuorum di broker Amazon MQ pada versi 3.13 atau lebih tinggi dengan bermigrasi di tempat.

1. Hentikan konsumen dan produsen.
2. Buat antrean kuorum sementara baru.
3. Konfigurasikan plug in Shovel untuk memindahkan pesan apa pun dari antrian cermin klasik lama ke antrian kuorum sementara yang baru. Setelah semua pesan dipindahkan ke antrian kuorum sementara, hapus Sekop.
4. Hapus antrian cermin klasik sumber. Kemudian, buat ulang antrian kuorum dengan nama dan binding yang sama dengan antrian cermin klasik sumber.
5. Buat Shovel baru untuk memindahkan pesan dari antrean kuorum sementara ke antrean kuorum baru.

Konfigurasi kebijakan untuk antrian kuorum untuk Amazon MQ untuk RabbitMQ

Anda dapat menambahkan konfigurasi kebijakan tertentu ke antrian kuorum untuk broker RabbitMQ Anda di Amazon MQ.

Saat Anda membuat kebijakan untuk antrian kuorum, Anda harus melakukan hal berikut:

- Hapus semua atribut kebijakan yang dimulai denganha, sepertiha-mode,ha-params,ha-sync-mode,ha-sync-batch-size,ha-promote-on-shutdown, danha-promote-on-failure.
- Hapus queue-mode.
- Ubah overflow saat disetel ke reject-publish-dlx

> **Important**
>
> Amazon MQ untuk RabbitMQ menerapkan semua atau tidak ada atribut dalam kebijakan. Anda tidak dapat membuat kebijakan yang berlaku untuk antrian cermin klasik dan antrian kuorum. Jika Anda ingin kebijakan Anda hanya berlaku untuk antrian kuorum, Anda harus menyetel ke. --apply-to quorum_queues Jika Anda menggunakan antrian cermin klasik dan antrian kuorum, Anda harus membuat kebijakan terpisah dengan--apply-to: classic_queues serta kebijakan antrian kuorum.

Anda tidak perlu mengubah `AWS-DEFAULT` kebijakan karena kebijakan tersebut secara otomatis mengadopsi jenis antrian baru di parameter “berlaku untuk”. Untuk informasi selengkapnya tentang kebijakan default Amazon MQ untuk RabbitMQ, lihat. Mengkonfigurasi kebijakan operator

Praktik terbaik untuk antrian kuorum untuk Amazon MQ untuk RabbitMQ

Sebaiknya gunakan praktik terbaik berikut untuk meningkatkan kinerja saat bekerja dengan antrian kuorum.

Menangani pesan racun dengan menetapkan batas pengiriman

Pesan racun terjadi ketika pesan gagal dan dikirim ulang beberapa kali. Anda dapat menetapkan batas pengiriman pesan menggunakan argumen `delivery-limit` kebijakan untuk menghapus pesan yang dikirim ulang beberapa kali. Jika pesan dikirim ulang lebih dari batas pengiriman yang diizinkan, pesan tersebut kemudian dihapus dan dihapus oleh RabbitMQ. Saat Anda menetapkan batas pengiriman, pesan akan diminta ulang di dekat kepala antrian.

Prioritas pesan untuk antrian kuorum

Antrian kuorum tidak memiliki prioritas pesan. Jika Anda membutuhkan prioritas pesan, Anda harus membuat beberapa antrian kuorum. Untuk informasi selengkapnya tentang memprioritaskan pesan dengan beberapa antrian kuorum, lihat Prioritas pesan dalam dokumentasi RabbitMQ.

Menggunakan faktor replikasi default

Amazon MQ untuk RabbitMQ default ke faktor replikasi tiga (3) node untuk broker klaster menggunakan antrian kuorum. Jika Anda membuat perubahan`x-quorum-initial-group-size`, Amazon MQ akan default lagi ke faktor replikasi 3.

Amazon MQ untuk praktik terbaik RabbitMQ

Ikuti pedoman kesiapan produksi ini untuk memaksimalkan kinerja broker dan mengoptimalkan efisiensi throughput pesan saat bekerja dengan Amazon MQ untuk broker RabbitMQ.

> **Important**
>
> Saat ini, Amazon MQ tidak mendukung aliran, atau menggunakan logging terstruktur di JSON, diperkenalkan di RabbitMQ 3.9.x.

Topik

- Praktik terbaik untuk pengaturan broker dan manajemen koneksi di Amazon MQ untuk RabbitMQ
- Praktik terbaik untuk daya tahan dan keandalan pesan di Amazon MQ untuk RabbitMQ
- Praktik terbaik untuk pengoptimalan kinerja dan efisiensi di Amazon MQ untuk RabbitMQ
- Praktik terbaik untuk ketahanan dan pemantauan jaringan di Amazon MQ untuk RabbitMQ

Praktik terbaik untuk pengaturan broker dan manajemen koneksi di Amazon MQ untuk RabbitMQ

Pengaturan broker dan manajemen koneksi adalah langkah pertama dalam mencegah masalah dengan throughput pesan broker, pemanfaatan sumber daya, dan kemampuan untuk menangani beban kerja produksi. Saat membuat dan mengonfigurasi Amazon MQ untuk broker RabbitMQ, selesaikan praktik terbaik berikut untuk memilih jenis instans yang sesuai, mengelola koneksi secara efisien, dan mengonfigurasi pra-pengambilan pesan untuk memaksimalkan kinerja broker Anda.

> **Important**
>
> Amazon MQ untuk RabbitMQ tidak mendukung nama pengguna “tamu”, dan akan menghapus akun tamu default saat Anda membuat broker baru. Amazon MQ juga akan secara berkala menghapus akun yang dibuat pelanggan yang disebut “tamu”.

Langkah 1: Gunakan penerapan cluster

Untuk beban kerja produksi, sebaiknya gunakan penerapan klaster alih-alih pialang instans tunggal untuk memastikan ketersediaan dan ketahanan pesan yang tinggi. Penerapan cluster menghapus satu titik kegagalan dan memberikan toleransi kesalahan yang lebih baik.

Penerapan cluster terdiri dari tiga node broker RabbitMQ yang didistribusikan di tiga Availability Zone, menyediakan failover otomatis dan memastikan operasi berlanjut bahkan jika seluruh Availability Zone menjadi tidak tersedia. Amazon MQ secara otomatis mereplikasi pesan di semua node untuk memastikan ketersediaan selama kegagalan atau pemeliharaan node.

Penerapan klaster sangat penting untuk lingkungan produksi dan didukung oleh Perjanjian Tingkat Layanan Amazon MQ.

Untuk informasi selengkapnya, lihat Penerapan klaster di Amazon MQ untuk RabbitMQ.

Langkah 2: Pilih jenis instans broker yang benar

Throughput pesan dari jenis instans broker tergantung pada kasus penggunaan aplikasi Anda. `M7g.mediumseharusnya` hanya digunakan untuk menguji kinerja aplikasi. Menggunakan instance yang lebih kecil ini sebelum menggunakan instance yang lebih besar dalam produksi dapat meningkatkan kinerja aplikasi. Pada jenis instans `m7g.large` dan di atasnya, Anda dapat menggunakan penerapan klaster untuk ketersediaan tinggi dan daya tahan pesan. Jenis instans broker yang lebih besar dapat menangani tingkat produksi klien dan antrian, throughput tinggi, pesan dalam memori, dan pesan yang berlebihan.

Untuk informasi selengkapnya tentang memilih jenis instans yang benar, lihat Pedoman ukuran di Amazon MQ for RabbitMQ.

Langkah 3: Gunakan antrian kuorum

Antrian kuorum, dengan penerapan klaster, harus menjadi pilihan default untuk jenis antrian yang direplikasi di lingkungan produksi untuk broker RabbitMQ pada 3.13 ke atas. Antrian kuorum adalah tipe antrian modern yang direplikasi yang memberikan keandalan tinggi, throughput tinggi, dan latensi stabil.

Antrian kuorum menggunakan algoritma konsensus Raft untuk memberikan toleransi kesalahan yang lebih baik. Ketika node pemimpin menjadi tidak tersedia, antrian kuorum secara otomatis memilih pemimpin baru dengan suara mayoritas, memastikan pengiriman pesan berlanjut dengan gangguan minimal. Karena setiap node berada di Availability Zone yang berbeda, sistem pesan Anda tetap tersedia meskipun seluruh Availability Zone menjadi tidak tersedia untuk sementara.

Untuk mendeklarasikan antrian kuorum, atur header saat membuat antrian `x-queue-type` Andaquorum.

Untuk informasi selengkapnya tentang antrian kuorum, termasuk strategi migrasi dan praktik terbaik, lihat Antrian kuorum di Amazon MQ untuk RabbitMQ.

Langkah 4: Gunakan beberapa saluran

Untuk menghindari churn koneksi, gunakan beberapa saluran melalui satu koneksi. Aplikasi harus menghindari rasio koneksi 1:1 ke saluran. Kami merekomendasikan menggunakan satu koneksi untuk setiap proses, dan kemudian satu saluran untuk setiap utas. Hindari penggunaan saluran yang berlebihan untuk mencegah kebocoran saluran.

Praktik terbaik untuk daya tahan dan keandalan pesan di Amazon MQ untuk RabbitMQ

Sebelum memindahkan aplikasi Anda ke produksi, selesaikan praktik terbaik berikut untuk mencegah kehilangan pesan dan pemanfaatan sumber daya yang berlebihan.

Langkah 1: Gunakan pesan persisten dan antrian yang tahan lama

Pesan persisten dapat membantu melindungi daya tahan data dalam situasi di mana broker crash atau restart. Pesan persisten ditulis ke disk segera setelah pesan tiba. Tidak seperti antrean malas, pesan persisten di-cache dalam memori dan disk, kecuali lebih banyak memori diperlukan oleh broker. Dalam kasus ketika lebih banyak memori diperlukan, pesan dihapus dari memori oleh mekanisme broker RabbitMQ yang mengelola penyimpanan pesan ke disk, sering disebut sebagai lapisan persisten.

Untuk mengaktifkan persistensi pesan, Anda dapat menyatakan antrean sebagai `durable` dan mengatur mode pengiriman pesan ke `persistent`. Contoh berikut mendemonstrasikan penggunaan pustaka klien RabbitMQ Java untuk mendeklarasikan antrean yang tahan lama. Saat bekerja dengan AMQP 0-9-1, Anda dapat menandai pesan sebagai persisten dengan mengatur mode pengiriman “2".

```
boolean durable = true;
channel.queueDeclare("my_queue", durable, false, false, null);
```

Setelah mengonfigurasi antrean sebagai tahan lama, Anda dapat mengirim pesan persisten ke antrean dengan mengatur `MessageProperties` ke `PERSISTENT_TEXT_PLAIN` seperti yang ditampilkan dalam contoh berikut.

```
import com.rabbitmq.client.MessageProperties;

channel.basicPublish("", "my_queue",
            MessageProperties.PERSISTENT_TEXT_PLAIN,
            message.getBytes());
```

Langkah 2: Konfigurasikan konfirmasi penerbit dan pengakuan pengiriman konsumen

Proses konfirmasi pesan telah dikirim ke broker dikenal sebagai konfirmasi penerbit. Publisher mengonfirmasi membiarkan aplikasi Anda tahu kapan pesan telah disimpan dengan andal. Konfirmasi penerbit juga dapat membantu mengontrol tingkat pesan yang disimpan ke broker. Tanpa

konfirmasi penerbit, tidak ada konfirmasi bahwa pesan diproses dengan sukses, dan broker Anda dapat menjatuhkan pesan yang tidak dapat diproses.

Demikian pula, ketika aplikasi klien mengirimkan konfirmasi pengiriman dan konsumsi pesan kembali ke broker, itu dikenal sebagai pengakuan pengiriman konsumen. Konfirmasi dan pengakuan sangat penting untuk memastikan keamanan data saat bekerja dengan broker RabbitMQ.

Pengakuan pengiriman konsumen biasanya dikonfigurasi pada aplikasi klien. Saat bekerja dengan AMQP 0-9-1, pengakuan dapat diaktifkan dengan mengonfigurasi metode. `basic.consume` Klien AMQP 0-9-1 juga dapat mengonfigurasi konfirmasi penerbit dengan mengirimkan metode. `confirm.select`

Biasanya, pengakuan pengiriman diaktifkan di saluran. Misalnya, ketika bekerja dengan pustaka klien RabbitMQ Java, Anda dapat menggunakan `Channel#basicAck` untuk menyiapkan yang pengakuan positif `basic.ack` sederhana seperti yang ditampilkan dalam contoh berikut.

```
// this example assumes an existing channel instance

boolean autoAck = false;
channel.basicConsume(queueName, autoAck, "a-consumer-tag",
     new DefaultConsumer(channel) {
         @Override
         public void handleDelivery(String consumerTag,
                                    Envelope envelope,
                                    AMQP.BasicProperties properties,
                                    byte[] body)
             throws IOException
         {
             long deliveryTag = envelope.getDeliveryTag();
             // positively acknowledge a single delivery, the message will
             // be discarded
             channel.basicAck(deliveryTag, false);
         }
     });
```

> **Note**
>
> Pesan yang tidak diakui harus di-cache dalam memori. Anda dapat membatasi jumlah pesan yang diambil sebelumnya oleh konsumen dengan mengonfigurasi pengaturan pra-pengambilan untuk aplikasi klien.

Anda dapat mengonfigurasi `consumer_timeout` untuk mendeteksi ketika konsumen tidak mengakui pengiriman. Jika konsumen tidak mengirimkan pengakuan dalam nilai batas waktu, saluran akan ditutup, dan Anda akan menerima a. `PRECONDITION_FAILED` Untuk mendiagnosis kesalahan, gunakan UpdateConfigurationAPI untuk meningkatkan `consumer_timeout` nilai.

Langkah 3: Jaga antrian pendek

Dalam penerapan cluster, antrian dengan sejumlah besar pesan dapat menyebabkan pemanfaatan sumber daya yang berlebihan. Ketika broker dimanfaatkan secara berlebihan, me-reboot Amazon MQ untuk broker RabbitMQ dapat menyebabkan penurunan kinerja lebih lanjut. Jika reboot, broker yang terlalu banyak digunakan mungkin menjadi tidak responsif di negara bagian. `REBOOT_IN_PROGRESS`

Selama jendela pemeliharaan, Amazon MQ melakukan semua pekerjaan pemeliharaan, satu simpul pada satu waktu, untuk memastikan bahwa broker tetap operasional. Akibatnya, antrian mungkin perlu disinkronkan karena setiap simpul melanjutkan operasi. Selama sinkronisasi, pesan yang perlu direplikasi ke cermin dimuat ke dalam memori dari volume Amazon Elastic Block Store (Amazon EBS) yang sesuai untuk diproses dalam batch. Memproses pesan dalam batch memungkinkan antrean menyinkronkan lebih cepat.

Jika antrean dibuat tetap pendek dan pesan berukuran kecil, antrean berhasil disinkronkan dan melanjutkan operasi seperti yang diharapkan. Namun, jika jumlah data dalam batch mendekati batas memori simpul, simpul memicu alarm memori tinggi, menjeda sinkronisasi antrean. Anda dapat mengonfirmasi penggunaan memori dengan membandingkan metrik node `RabbitMemUsed` dan `RabbitMqMemLimit` broker di CloudWatch. Sinkronisasi tidak dapat diselesaikan hingga pesan dikonsumsi atau dihapus, atau jumlah pesan dalam batch berkurang.

Jika sinkronisasi antrian dijeda untuk penerapan klaster, sebaiknya gunakan atau hapus pesan untuk menurunkan jumlah pesan dalam antrian. Setelah kedalaman antrian berkurang dan sinkronisasi antrian selesai, status broker akan berubah menjadi. `RUNNING` Untuk menyelesaikan sinkronisasi antrian yang dijeda, Anda juga dapat menerapkan kebijakan untuk mengurangi ukuran batch sinkronisasi antrian.

Anda juga dapat menentukan kebijakan penghapusan otomatis dan TTL untuk secara proaktif mengurangi penggunaan sumber daya, serta menjaga NACKs dari konsumen seminimal mungkin. Requeueing pesan pada broker adalah CPU intensif sehingga jumlah yang tinggi NACKs dapat mempengaruhi kinerja broker.

Praktik terbaik untuk pengoptimalan kinerja dan efisiensi di Amazon MQ untuk RabbitMQ

Anda dapat mengoptimalkan Amazon MQ Anda untuk kinerja broker RabbitMQ dengan memaksimalkan throughput, meminimalkan latensi, dan memastikan pemanfaatan sumber daya yang efisien. Selesaikan praktik terbaik berikut untuk mengoptimalkan kinerja aplikasi Anda.

Langkah 1: Simpan ukuran pesan di bawah 1 MB

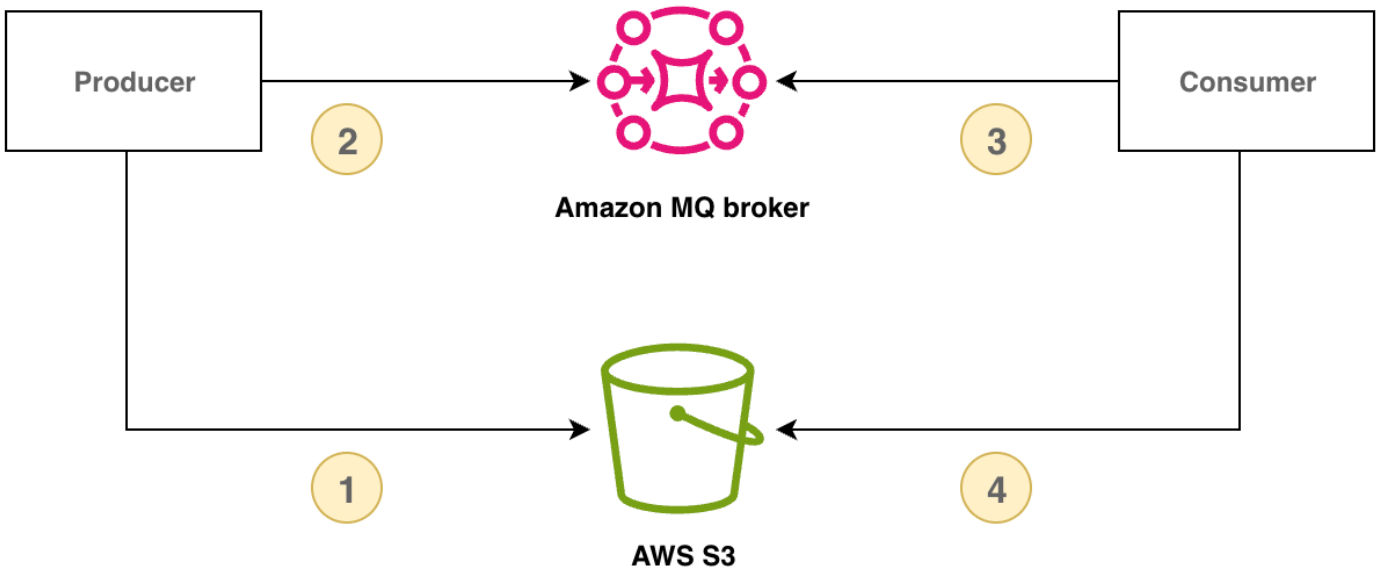
Sebaiknya simpan pesan di bawah 1 Megabyte (MB) untuk kinerja dan keandalan yang optimal.

RabbitMQ 3.13 mendukung ukuran pesan hingga 128 MB secara default, tetapi pesan besar dapat memicu alarm memori yang tidak terduga yang memblokir penerbitan dan berpotensi menciptakan tekanan memori tinggi saat mereplikasi pesan di seluruh node. Pesan yang terlalu besar juga dapat memengaruhi proses restart dan pemulihan broker, yang meningkatkan risiko terhadap kontinuitas layanan dan dapat menyebabkan penurunan kinerja.

Simpan dan ambil muatan besar menggunakan pola pemeriksaan klaim

Untuk mengelola pesan besar, Anda dapat menerapkan pola pemeriksaan klaim dengan menyimpan muatan pesan di penyimpanan eksternal dan hanya mengirim pengidentifikasi referensi payload melalui RabbitMQ. Konsumen menggunakan pengidentifikasi referensi payload untuk mengambil dan memproses pesan besar.

Diagram berikut menunjukkan cara menggunakan Amazon MQ untuk RabbitMQ dan Amazon S3 untuk menerapkan pola pemeriksaan klaim.



Contoh berikut menunjukkan pola ini menggunakan Amazon MQ, SDK for Java AWS 2.x, dan Amazon S3:

1. Pertama, tentukan kelas Pesan yang akan menampung pengenal referensi Amazon S3.

```
class Message {
    // Other data fields of the message...

    public String s3Key;
    public String s3Bucket;
}
```

2. Buat metode penerbit yang menyimpan muatan di Amazon S3 dan mengirim pesan referensi melalui RabbitMQ.

```
public void publishPayload() {
    // Store the payload in S3.
    String payload = PAYLOAD;
    String prefix = S3_KEY_PREFIX;
    String s3Key = prefix + "/" + UUID.randomUUID();
    s3Client.putObject(PutObjectRequest.builder()
        .bucket(S3_BUCKET).key(s3Key).build(),
        RequestBody.fromString(payload));

    // Send the reference through RabbitMQ.
    Message message = new Message();
```

```
    message.s3Key = s3Key;
    message.s3Bucket = S3_BUCKET;
    // Assign values to other fields in your message instance.

    publishMessage(message);
}
```

3. Menerapkan metode konsumen yang mengambil payload dari Amazon S3, memproses payload, dan menghapus objek Amazon S3.

```
public void consumeMessage(Message message) {
    // Retrieve the payload from S3.
    String payload = s3Client.getObjectAsBytes(GetObjectRequest.builder()
        .bucket(message.s3Bucket).key(message.s3Key).build())
        .asUtf8String();

    // Process the complete message.
    processPayload(message, payload);

    // Delete the S3 object.
    s3Client.deleteObject(DeleteObjectRequest.builder()
        .bucket(message.s3Bucket).key(message.s3Key).build());
}
```

Langkah 2: Gunakan **`basic.consume`** dan konsumen berumur panjang

Menggunakan `basic.consume` dengan konsumen berumur panjang lebih efisien daripada polling untuk menggunakan pesan individu. `basic.get` Untuk informasi selengkapnya, lihat Polling untuk pesan individual.

Langkah 3: Konfigurasikan pra-pengambilan

Anda dapat menggunakan nilai pra-pengambilan RabbitMQ untuk mengoptimalkan cara konsumen mengonsumsi pesan. RabbitMQ mengimplementasikan mekanisme pra-pengambilan saluran yang disediakan oleh AMQP 0-9-1 dengan menerapkan jumlah pra-pengambilan untuk konsumen yang bertentangan dengan saluran. Nilai pra-pengambilan digunakan untuk menentukan jumlah pesan yang dikirim ke konsumen pada waktu tertentu. Secara default, RabbitMQ menetapkan ukuran buffer yang tidak terbatas untuk aplikasi klien.

Ada berbagai faktor yang perlu dipertimbangkan saat menetapkan jumlah pra-pengambilan untuk konsumen RabbitMQ. Pertama, pertimbangkan lingkungan dan konfigurasi konsumen Anda. Karena

konsumen perlu menyimpan semua pesan dalam memori saat pesan sedang diproses, nilai pra-pengambilan yang tinggi dapat memiliki dampak negatif pada performa konsumen, dan di beberapa kasus, membuat konsumen berpotensi merusak semuanya. Demikian pula, broker RabbitMQ sendiri menyimpan semua pesan yang dikirimkannya dalam cache dalam memori sampai menerima pengakuan konsumen. Nilai pra-pengambilan yang tinggi dapat menyebabkan server RabbitMQ Anda kehabisan memori dengan cepat jika pengakuan otomatis tidak dikonfigurasi untuk konsumen, dan jika konsumen mengambil waktu yang relatif lama untuk memproses pesan.

Dengan pertimbangan di atas, kami rekomendasikan Anda untuk selalu menetapkan nilai pra-pengambilan agar terhindar dari situasi ketika broker RabbitMQ atau konsumen kehabisan memori karena sejumlah besar pesan yang belum diproses, atau tidak diakui. Jika perlu mengoptimalkan broker untuk memproses pesan dalam volume besar, Anda dapat menguji broker dan konsumen menggunakan berbagai jumlah pra-pengambilan untuk menentukan nilai titik ketika overhead jaringan menjadi sangat tidak signifikan dibandingkan dengan waktu yang dibutuhkan konsumen untuk memproses pesan.

> **Note**
>
> - Jika aplikasi klien Anda telah dikonfigurasi untuk secara otomatis mengakui pengiriman pesan ke konsumen, menetapkan nilai pra-pengambilan tidak akan berpengaruh.
> - Semua pesan pra-pengambilan dihapus dari antrean.

Contoh berikut mendemonstrasikan cara menentukan nilai pra-pengambilan `10` untuk konsumen tunggal menggunakan pustaka klien RabbitMQ Java.

```
ConnectionFactory factory = new ConnectionFactory();

Connection connection = factory.newConnection();
Channel channel = connection.createChannel();

channel.basicQos(10, false);

QueueingConsumer consumer = new QueueingConsumer(channel);
channel.basicConsume("my_queue", false, consumer);
```

> **Note**
>
> Dalam pustaka klien RabbitMQ Java, nilai default untuk bendera `global`diatur ke `false`, sehingga contoh di atas dapat ditulis hanya sebagai `channel.basicQos(10)`.

Langkah 4: Gunakan Seledri 5.5 atau lebih baru dengan antrian kuorum

Python Celery, sistem antrian tugas terdistribusi, dapat menghasilkan banyak pesan non-kritis saat mengalami beban tugas yang tinggi. Aktivitas broker tambahan ini dapat memicu the section called "`RABBITMQ_MEMORY_ALARM`" dan menyebabkan tidak tersedianya broker. Untuk mengurangi kemungkinan memicu alarm memori, lakukan hal berikut:

Untuk semua versi Seledri

1. Matikan `task_create_missing_queues`untuk mengurangi churn antrian.
2. Kemudian, matikan `worker_enable_remote_control` untuk menghentikan pembuatan `celery@...pidbox` antrian dinamis. Ini akan mengurangi churn antrian pada broker.

   ```
   worker_enable_remote_control = false
   ```
3. Untuk lebih mengurangi aktivitas pesan non-kritis, matikan Seledri worker-send-task-eventsdengan tidak menyertakan `-E` atau `--task-events` menandai saat memulai aplikasi Seledri Anda.
4. Mulai aplikasi Seledri Anda menggunakan parameter berikut:

   ```
   celery -A app_name worker --without-heartbeat --without-gossip --without-mingle
   ```

Untuk Seledri versi 5.5 dan di atas

1. Tingkatkan ke Celery versi 5.5, versi minimum yang mendukung antrian kuorum, atau versi yang lebih baru. Untuk memeriksa versi Seledri apa yang Anda gunakan, gunakan`celery --version`. Untuk informasi lebih lanjut tentang antrian kuorum, lihat. the section called "Antrian kuorum"
2. Setelah memutakhirkan ke Celery 5.5 atau yang lebih baru, konfigurasikan `task_default_queue_type` ke "kuorum".
3. Kemudian, Anda juga harus mengaktifkan Publikasikan Konfirmasi di Opsi Transportasi Broker:

```
broker_transport_options = {"confirm_publish": True}
```

Praktik terbaik untuk ketahanan dan pemantauan jaringan di Amazon MQ untuk RabbitMQ

Ketahanan jaringan dan metrik pialang pemantauan sangat penting untuk memelihara aplikasi perpesanan yang andal. Selesaikan praktik terbaik berikut untuk menerapkan mekanisme pemulihan otomatis dan strategi pemantauan sumber daya.

Langkah 1: Secara otomatis pulih dari kegagalan jaringan

Kami merekomendasikan untuk selalu mengaktifkan pemulihan jaringan otomatis guna mencegah waktu henti yang signifikan ketika koneksi klien ke node RabbitMQ gagal. Pustaka klien RabbitMQ Java mendukung pemulihan jaringan otomatis secara default, dimulai dari versi `4.0.0`.

Pemulihan koneksi otomatis dipicu jika pengecualian yang tidak tertangani dilemparkan ke I/O loop koneksi, jika batas waktu operasi baca soket terdeteksi, atau jika server melewatkan detak jantung.

Dalam kasus ketika koneksi awal antara klien dan node RabbitMQ gagal, pemulihan otomatis tidak akan dipicu. Kami merekomendasikan Anda menulis kode aplikasi untuk memperhitungkan kegagalan koneksi awal dengan mencoba ulang koneksi. Contoh berikut mendemonstrasikan percobaan ulang kegagalan jaringan awal menggunakan pustaka klien RabbitMQ Java.

```
ConnectionFactory factory = new ConnectionFactory();
// enable automatic recovery if using RabbitMQ Java client library prior to version
 4.0.0.
factory.setAutomaticRecoveryEnabled(true);
// configure various connection settings

try {
  Connection conn = factory.newConnection();
} catch (java.net.ConnectException e) {
  Thread.sleep(5000);
  // apply retry logic
}
```

> **Note**
>
> Jika aplikasi menutup koneksi menggunakan metode `Connection.Close`, pemulihan jaringan otomatis tidak akan diaktifkan atau dipicu.

Langkah 2: Pantau metrik dan alarm broker

Kami merekomendasikan pemantauan CloudWatch metrik dan alarm secara teratur untuk Amazon MQ Anda untuk broker RabbitMQ untuk mengidentifikasi dan mengatasi potensi masalah sebelum memengaruhi aplikasi perpesanan Anda. Pemantauan proaktif sangat penting untuk menjaga aplikasi pesan yang tangguh dan memastikan kinerja yang optimal.

Amazon MQ untuk RabbitMQ menerbitkan metrik CloudWatch yang memberikan wawasan tentang kinerja broker, pemanfaatan sumber daya, dan alur pesan. Metrik utama untuk memantau termasuk penggunaan memori dan penggunaan disk. Anda dapat mengatur CloudWatch alarm ketika broker Anda mendekati batas sumber daya atau mengalami penurunan kinerja.

Pantau metrik penting berikut:

`RabbitMQMemUsed` dan **`RabbitMQMemLimit`**

Pantau penggunaan memori untuk mencegah alarm memori yang dapat memblokir penerbitan pesan.

`RabbitMQDiskFree` dan **`RabbitMQDiskFreeLimit`**

Pantau penggunaan disk untuk menghindari masalah ruang disk yang dapat menyebabkan kegagalan broker.

Untuk penerapan klaster, pantau juga metrik khusus node untuk mengidentifikasi masalah spesifik node.

> **Note**
>
> Untuk informasi selengkapnya tentang cara mencegah alarm memori tinggi, lihat Alamat dan mencegah alarm memori tinggi.

Tutorial RabbitMQ

Tutorial berikut menunjukkan cara mengonfigurasi dan menggunakan RabbitMQ di Amazon MQ. Untuk mempelajari lebih lanjut tentang bekerja dengan pustaka klien yang didukung dalam berbagai bahasa pemrograman seperti Node.js, Python, .NET, dan lainnya, lihat Tutorial RabbitMQ dalam Panduan Memulai RabbitMQ.

Topik

- Mengedit preferensi broker
- Menggunakan Python Pika dengan Amazon MQ untuk RabbitMQ
- Menyelesaikan sinkronisasi antrean RabbitMQ yang dijeda
- Mengurangi jumlah koneksi dan saluran
- Langkah 2: Hubungkan aplikasi berbasis JVM ke broker Anda
- Langkah 3: (Opsional) Connect ke AWS Lambda fungsi
- Menggunakan otentikasi OAuth 2.0 dan otorisasi untuk Amazon MQ untuk RabbitMQ
- Menggunakan otentikasi dan otorisasi IAM untuk Amazon MQ untuk RabbitMQ
- Menggunakan otentikasi dan otorisasi LDAP untuk Amazon MQ untuk RabbitMQ
- Menggunakan otentikasi HTTP dan otorisasi untuk Amazon MQ untuk RabbitMQ
- Menggunakan otentikasi sertifikat SSL untuk Amazon MQ untuk RabbitMQ
- Menggunakan mTL untuk AMQP dan endpoint manajemen
- Menghubungkan aplikasi JMS Anda

Mengedit preferensi broker

Anda dapat mengedit preferensi broker Anda, seperti mengaktifkan atau menonaktifkan CloudWatch log menggunakan. Konsol Manajemen AWS

Mengedit opsi broker RabbitMQ

1. Masuk ke konsol Amazon MQ.
2. Dari daftar broker, pilih broker Anda (misalnya, MyBroker) dan kemudian pilih Edit.
3. Pada *MyBroker* halaman Edit, di bagian Spesifikasi, pilih versi mesin Broker atau jenis Instance Broker.

4. Di bagian CloudWatch Log, klik tombol toggle untuk mengaktifkan atau menonaktifkan log umum. Tidak ada langkah lain yang diperlukan.

   > **Note**
   >
   > - Untuk broker RabbitMQ, Amazon MQ secara otomatis menggunakan Service-Linked Role (SLR) untuk mempublikasikan log umum ke. CloudWatch Untuk informasi selengkapnya, lihat the section called “Menggunakan peran yang terhubung dengan layanan”
   > - Amazon MQ tidak mendukung pencatatan audit untuk broker RabbitMQ.

5. Di bagian Pemeliharaan, konfigurasikan jadwal pemeliharaan broker Anda:

   Untuk meningkatkan broker ke versi baru saat AWS merilisnya, pilih Aktifkan peningkatan versi minor otomatis. Peningkatan otomatis terjadi selama jendela pemeliharaan yang ditentukan oleh hari dalam seminggu, waktu dalam sehari (dalam format 24 jam), dan zona waktu (UTC secara default).

6. Pilih Perubahan jadwal.

   > **Note**
   >
   > Jika Anda hanya memilih Aktifkan peningkatan versi minor otomatis, tombol berubah menjadi Simpan karena boot ulang broker tidak diperlukan.

   Preferensi Anda diterapkan pada broker Anda pada waktu yang ditentukan.

Menggunakan Python Pika dengan Amazon MQ untuk RabbitMQ

Tutorial berikut menunjukkan bagaimana Anda dapat mengatur klien Python Pika dengan TLS dikonfigurasi untuk terhubung ke Amazon MQ untuk broker RabbitMQ. Pika adalah implementasi Python dari protokol AMQP 0-9-1 untuk RabbitMQ. Tutorial ini memandu Anda melalui instalasi Pika, mendeklarasikan antrian, menyiapkan penerbit untuk mengirim pesan ke pertukaran default broker, dan menyiapkan konsumen untuk menerima pesan dari antrian.

Topik

- Prasyarat
- Izin
- Langkah satu: Buat klien Python Pika dasar
- Langkah kedua: Buat penerbit dan kirim pesan
- Langkah ketiga: Buat konsumen dan terima pesan
- Langkah empat: (Opsional) Siapkan loop acara dan gunakan pesan
- Apa selanjutnya?

Prasyarat

Untuk menyelesaikan langkah-langkah dalam tutorial ini, Anda memerlukan prasyarat berikut:

- Amazon MQ untuk broker RabbitMQ. Untuk informasi selengkapnya, lihat Membuat Amazon MQ untuk broker RabbitMQ.
- Python 3 diinstal untuk sistem operasi Anda.
- Pika diinstal menggunakan Python`pip`. Untuk menginstal Pika, buka jendela terminal baru dan jalankan yang berikut ini.

```
$ python3 -m pip install pika
```

Izin

Untuk tutorial ini, Anda memerlukan setidaknya satu Amazon MQ untuk pengguna broker RabbitMQ dengan izin untuk menulis ke, dan membaca dari, vhost. Tabel berikut menjelaskan izin minimum yang diperlukan sebagai pola ekspresi reguler (regexp).

Tag	Konfigurasikan regexp	Tulis regexp	Baca regexp
none		`.*`	`.*`

Izin pengguna yang tercantum hanya memberikan izin baca dan tulis kepada pengguna, tanpa memberikan akses ke plugin manajemen untuk melakukan operasi administratif pada broker. Anda dapat membatasi izin lebih lanjut dengan menyediakan pola regexp yang membatasi akses pengguna ke antrian tertentu. Misalnya, jika Anda mengubah pola regexp baca menjadi`^[hello`

`world].*`, pengguna hanya akan memiliki izin untuk membaca dari antrian yang dimulai dengan. `hello world`

Untuk informasi selengkapnya tentang cara membuat pengguna RabbitMQ serta mengelola tanda dan izin pengguna, lihat Amazon MQ untuk pengguna broker RabbitMQ.

Langkah satu: Buat klien Python Pika dasar

Untuk membuat kelas basis klien Python Pika yang mendefinisikan konstruktor dan menyediakan konteks SSL yang diperlukan untuk konfigurasi TLS saat berinteraksi dengan Amazon MQ untuk broker RabbitMQ, lakukan hal berikut.

1. Buka jendela terminal baru, buat direktori baru untuk proyek Anda, dan arahkan ke direktori.

```
$ mkdir pika-tutorial
$ cd pika-tutorial
```

2. Buat file baru,`basicClient.py`, yang berisi kode Python berikut.

```
import ssl
import pika

class BasicPikaClient:

    def __init__(self, rabbitmq_broker_id, rabbitmq_user, rabbitmq_password,
 region):

        # SSL Context for TLS configuration of Amazon MQ for RabbitMQ
        ssl_context = ssl.SSLContext(ssl.PROTOCOL_TLSv1_2)
        ssl_context.set_ciphers('ECDHE+AESGCM:!ECDSA')

        url = f"amqps://{rabbitmq_user}:
{rabbitmq_password}@{rabbitmq_broker_id}.mq.{region}.amazonaws.com:5671"
        parameters = pika.URLParameters(url)
        parameters.ssl_options = pika.SSLOptions(context=ssl_context)

        self.connection = pika.BlockingConnection(parameters)
        self.channel = self.connection.channel()
```

Anda sekarang dapat menentukan kelas tambahan untuk penerbit dan konsumen yang mewarisi dari. `BasicPikaClient`

Langkah kedua: Buat penerbit dan kirim pesan

Untuk membuat penerbit yang mendeklarasikan antrian, dan mengirim satu pesan, lakukan hal berikut.

1. Salin isi contoh kode berikut, dan simpan secara lokal seperti `publisher.py` di direktori yang sama yang Anda buat pada langkah sebelumnya.

```
from basicClient import BasicPikaClient

class BasicMessageSender(BasicPikaClient):

    def declare_queue(self, queue_name):
        print(f"Trying to declare queue({queue_name})...")
        self.channel.queue_declare(queue=queue_name)

    def send_message(self, exchange, routing_key, body):
        channel = self.connection.channel()
        channel.basic_publish(exchange=exchange,
                              routing_key=routing_key,
                              body=body)
        print(f"Sent message. Exchange: {exchange}, Routing Key: {routing_key},
 Body: {body}")

    def close(self):
        self.channel.close()
        self.connection.close()

if __name__ == "__main__":

    # Initialize Basic Message Sender which creates a connection
    # and channel for sending messages.
    basic_message_sender = BasicMessageSender(
        "<broker-id>",
        "<username>",
        "<password>",
        "<region>"
    )

    # Declare a queue
    basic_message_sender.declare_queue("hello world queue")

    # Send a message to the queue.
```

```
    basic_message_sender.send_message(exchange="", routing_key="hello world queue",
body=b'Hello World!')

    # Close connections.
    basic_message_sender.close()
```

`BasicMessageSender`Kelas mewarisi dari `BasicPikaClient` dan mengimplementasikan metode tambahan untuk menghapus antrian, mengirim pesan ke antrian, dan menutup koneksi. Contoh kode merutekan pesan ke pertukaran default, dengan kunci routing sama dengan nama antrian.

2. Di bawah`if __name__ == "__main__":`, ganti parameter yang diteruskan ke pernyataan `BasicMessageSender` konstruktor dengan informasi berikut.

   - **<broker-id>** – ID unik yang dihasilkan Amazon MQ untuk broker. Anda dapat mengurai ID dari ARN broker. Misalnya, dengan ARN berikut, `arn:aws:mq:us-east-2:123456789012:broker:MyBroker:b-1234a5b6-78cd-901e-2fgh-3i45j6k178l9`, ID broker akan menjadi `b-1234a5b6-78cd-901e-2fgh-3i45j6k178l9`.
   - **<username>**— Nama pengguna untuk pengguna broker dengan izin yang cukup untuk menulis pesan ke broker.
   - **<password>**— Kata sandi untuk pengguna broker dengan izin yang cukup untuk menulis pesan ke broker.
   - **<region>**— AWS Wilayah tempat Anda membuat Amazon MQ untuk broker RabbitMQ. Misalnya, `us-west-2`.

3. Jalankan perintah berikut di direktori yang sama yang Anda buat`publisher.py`.

```
$ python3 publisher.py
```

Jika kode berjalan dengan sukses, Anda akan melihat output berikut di jendela terminal Anda.

```
Trying to declare queue(hello world queue)...
Sent message. Exchange: , Routing Key: hello world queue, Body: b'Hello World!'
```

Langkah ketiga: Buat konsumen dan terima pesan

Untuk membuat konsumen yang menerima satu pesan dari antrian, lakukan hal berikut.

1. Salin isi contoh kode berikut, dan simpan secara lokal seperti `consumer.py` di direktori yang sama.

```
from basicClient import BasicPikaClient

class BasicMessageReceiver(BasicPikaClient):

    def get_message(self, queue):
        method_frame, header_frame, body = self.channel.basic_get(queue)
        if method_frame:
            print(method_frame, header_frame, body)
            self.channel.basic_ack(method_frame.delivery_tag)
            return method_frame, header_frame, body
        else:
            print('No message returned')

    def close(self):
        self.channel.close()
        self.connection.close()


if __name__ == "__main__":

    # Create Basic Message Receiver which creates a connection
    # and channel for consuming messages.
    basic_message_receiver = BasicMessageReceiver(
        "<broker-id>",
        "<username>",
        "<password>",
        "<region>"
    )

    # Consume the message that was sent.
    basic_message_receiver.get_message("hello world queue")

    # Close connections.
    basic_message_receiver.close()
```

Mirip dengan penerbit yang Anda buat pada langkah sebelumnya, `BasicMessageReceiver` mewarisi dari `BasicPikaClient` dan mengimplementasikan metode tambahan untuk menerima satu pesan, dan menutup koneksi.

2. Di bawah `if __name__ == "__main__":` pernyataan itu, ganti parameter yang diteruskan ke `BasicMessageReceiver` konstruktor dengan informasi Anda.
3. Jalankan perintah berikut di direktori proyek Anda.

```
$ python3 consumer.py
```

Jika kode berjalan dengan sukses, Anda akan melihat isi pesan, dan header termasuk tombol routing, ditampilkan di jendela terminal Anda.

```
<Basic.GetOk(['delivery_tag=1', 'exchange=', 'message_count=0',
 'redelivered=False', 'routing_key=hello world queue'])> <BasicProperties> b'Hello
 World!'
```

Langkah empat: (Opsional) Siapkan loop acara dan gunakan pesan

Untuk menggunakan beberapa pesan dari antrian, gunakan [basic_consume](#)metode Pika dan fungsi callback seperti yang ditunjukkan pada berikut

1. Dalam`consumer.py`, tambahkan definisi metode berikut ke `BasicMessageReceiver` kelas.

```
def consume_messages(self, queue):
    def callback(ch, method, properties, body):
        print(" [x] Received %r" % body)

    self.channel.basic_consume(queue=queue, on_message_callback=callback,
 auto_ack=True)

    print(' [*] Waiting for messages. To exit press CTRL+C')
    self.channel.start_consuming()
```

2. Di`consumer.py`, di bawah`if __name__ == "__main__":`, panggil `consume_messages` metode yang Anda tentukan di langkah sebelumnya.

```
if __name__ == "__main__":

    # Create Basic Message Receiver which creates a connection and channel for
 consuming messages.
    basic_message_receiver = BasicMessageReceiver(
        "<broker-id>",
```

```
        "<username>",
        "<password>",
        "<region>"
    )

    # Consume the message that was sent.
    # basic_message_receiver.get_message("hello world queue")

    # Consume multiple messages in an event loop.
    basic_message_receiver.consume_messages("hello world queue")

    # Close connections.
    basic_message_receiver.close()
```

3. Jalankan `consumer.py` lagi, dan jika berhasil, pesan antrian akan ditampilkan di jendela terminal Anda.

```
[*] Waiting for messages. To exit press CTRL+C
[x] Received b'Hello World!'
[x] Received b'Hello World!'
...
```

Apa selanjutnya?

- Untuk informasi selengkapnya tentang pustaka klien RabbitMQ lain yang didukung, lihat Dokumentasi Klien RabbitMQ di situs web RabbitMQ.

Menyelesaikan sinkronisasi antrean RabbitMQ yang dijeda

Dalam deployment klaster Amazon MQ for RabbitMQ, pesan yang dipublikasikan untuk setiap antrean direplikasi di tiga simpul broker. Replikasi ini, disebut sebagai pencerminan, menyediakan ketersediaan tinggi (HA) untuk broker RabbitMQ. Antrean dalam deployment klaster terdiri dari replika utama di satu simpul dan satu atau lebih cermin. Setiap operasi diterapkan ke antrean cermin, termasuk pesan penambahan antrean, terlebuh dahulu diterapkan ke antrean utama lalu direplikasi di seluruh cermin.

Misalnya, pertimbangkan antrean yang dicerminkan direplikasi di tiga simpul: simpul utama (`main`) dan dua cermin (`mirror-1` dan `mirror-2`). Jika semua pesan dalam antrean cermin ini berhasil disebarkan ke semua cermin, antrean akan disinkronkan. Jika simpul (`mirror-1`) menjadi tidak

tersedia selama interval waktu tertentu, antrean tetap operasional dan dapat terus menambahkan antrean pesan. Namun, untuk menyinkronkan antrean, pesan dipublikasikan ke `main`, sementara `mirror-1` tidak tersedia dan harus direplikasi ke `mirror-1`.

Untuk informasi selengkapnya tentang pencerminan, lihat Antrean Klasik yang Dicerminkan di situs RabbitMQ.

Pemeliharaan dan sinkronisasi antrian

Selama jendela pemeliharaan, Amazon MQ melakukan semua pekerjaan pemeliharaan, satu simpul pada satu waktu, untuk memastikan bahwa broker tetap operasional. Akibatnya, antrian mungkin perlu disinkronkan karena setiap simpul melanjutkan operasi. Selama sinkronisasi, pesan yang perlu direplikasi ke cermin dimuat ke dalam memori dari volume Amazon Elastic Block Store (Amazon EBS) yang sesuai untuk diproses dalam batch. Memproses pesan dalam batch memungkinkan antrean menyinkronkan lebih cepat.

Jika antrean dibuat tetap pendek dan pesan berukuran kecil, antrean berhasil disinkronkan dan melanjutkan operasi seperti yang diharapkan. Namun, jika jumlah data dalam batch mendekati batas memori simpul, simpul memicu alarm memori tinggi, menjeda sinkronisasi antrean. Anda dapat mengonfirmasi penggunaan memori dengan membandingkan metrik node `RabbitMemUsed` dan `RabbitMqMemLimit` broker di CloudWatch. Sinkronisasi tidak dapat diselesaikan hingga pesan dikonsumsi atau dihapus, atau jumlah pesan dalam batch berkurang.

> **Note**
>
> Mengurangi ukuran batch sinkronisasi antrean dapat mengakibatkan jumlah transaksi replikasi yang lebih tinggi.

Untuk mengatasi sinkronisasi antrean yang dijeda, ikuti langkah-langkah dalam tutorial ini, yang menunjukkan cara menerapkan kebijakan `ha-sync-batch-size` dan memulai ulang sinkronisasi antrean.

Topik

- Prasyarat
- Langkah 1: Menerapkan kebijakan ha-sync-batch-size
- Langkah 2: Memulai ulang sinkronisasi antrean
- Langkah selanjutnya

- Sumber daya terkait

Prasyarat

Untuk tutorial ini, Anda harus memiliki pengguna broker Amazon MQ for RabbitMQ dengan izin administrator. Anda dapat menggunakan pengguna administrator yang dibuat ketika pertama kali membuat broker, atau pengguna lain yang mungkin telah Anda buat sesudahnya. Tabel berikut menyediakan tanda pengguna administrator yang diperlukan dan izin sebagai pola ekspresi reguler (regexp).

Tanda	Baca regexp	Konfigurasikan regexp	Tulis regexp
`administrator`	`.*`	`.*`	`.*`

Untuk informasi selengkapnya tentang cara membuat pengguna RabbitMQ serta mengelola tanda dan izin pengguna, lihat Amazon MQ untuk pengguna broker RabbitMQ.

Langkah 1: Menerapkan kebijakan **`ha-sync-batch-size`**

Prosedur berikut mendemonstrasikan penambahan kebijakan yang berlaku untuk semua antrean yang dibuat pada broker. Anda dapat menggunakan konsol web RabbitMQ atau API manajemen RabbitMQ. Untuk informasi selengkapnya, lihat Plugin Manajemen di situs web RabbitMQ.

Untuk menerapkan kebijakan **`ha-sync-batch-size`** menggunakan konsol web RabbitMQ

1. Masuk ke konsol Amazon MQ.
2. Di panel navigasi kiri, pilih Broker.
3. Dari daftar broker, pilih nama broker yang ingin Anda terapkan kebijakan baru.
4. Di halaman broker, pada bagian Koneksi, pilih URL konsol web RabbitMQ. Konsol web RabbitMQ terbuka di tab browser atau jendela baru.
5. Masuk ke konsol web RabbitMQ dengan kredenal masuk administrator broker Anda.
6. Di konsol web RabbitMQ, di bagian atas halaman, pilih Admin.
7. Di halaman Admin, di panel navigasi kanan, pilih Kebijakan.
8. Di halaman Kebijakan, Anda dapat melihat daftar Kebijakan pengguna broker saat ini. Di bawah Kebijakan pengguna, perluas Tambahkan / perbarui kebijakan.

> **Note**
>
> Secara default, klaster Amazon MQ for RabbitMQ dibuat dengan kebijakan broker awal bernama `ha-all-AWS-OWNED-DO-NOT-DELETE`. Amazon MQ mengelola kebijakan ini untuk memastikan bahwa setiap antrean pada broker direplikasi ke ketiga simpul dan antrean tersebut disinkronkan secara otomatis.

9. Untuk membuat kebijakan broker baru, di bawah Tambahkan / perbarui kebijakan, lakukan hal berikut:

   a. Untuk Nama, masukkan nama untuk kebijakan Anda, misalnya **`batch-size-policy`**.

   b. Untuk Pola, masukkan pola regexp **`.*`** sehingga kebijakan cocok dengan semua antrean pada broker.

   c. Untuk Terapkan ke, pilih Pertukaran dan antrean dari daftar dropdown.

   d. Untuk Prioritas, masukkan integer yang lebih besar dari semua kebijakan lain yang diterapkan ke vhost. Anda dapat menerapkan satu set definisi kebijakan ke antrean dan pertukaran RabbitMQ pada waktu tertentu. RabbitMQ memilih kebijakan yang cocok dengan nilai prioritas tertinggi. Untuk informasi selengkapnya tentang prioritas kebijakan dan cara menggabungkan kebijakan, lihat Kebijakan dalam Dokumentasi Server RabbitMQ.

   e. Untuk Definisi, tambahkan pasangan nilai kunci berikut:

   - **`ha-sync-batch-size=`***`100`*. Pilih Nomor dari daftar dropdown.

     > **Note**
     >
     > Anda mungkin perlu menyesuaikan dan mengalibrasi nilai `ha-sync-batch-size` berdasarkan jumlah dan ukuran pesan yang tidak disinkronkan dalam antrean.

   - **`ha-mode=all`**. Pilih String dari daftar dropdown.

     > **Important**
     >
     > Definisi `ha-mode` diperlukan untuk semua kebijakan terkait HA. Menghilangkan hasilnya mengakibatkan kegagalan validasi.

   - **`ha-sync-mode=automatic`**. Pilih String dari daftar dropdown.

> **Note**
>
> Definisi `ha-sync-mode` diperlukan untuk semua kebijakan kustom. Jika dihilangkan, Amazon MQ secara otomatis menambahkan definisi.

f. Pilih Buat / perbarui kebijakan.

10. Konfirmasi bahwa kebijakan baru muncul dalam daftar Kebijakan pengguna.

Untuk menerapkan kebijakan **`ha-sync-batch-size`** menggunakan API manajemen RabbitMQ

1. Masuk ke konsol Amazon MQ.
2. Di panel navigasi kiri, pilih Broker.
3. Dari daftar broker, pilih nama broker yang ingin Anda terapkan kebijakan baru.
4. Di halaman broker, pada bagian Koneksi, catat URL konsol web RabbitMQ. Ini adalah titik akhir broker yang Anda gunakan dalam permintaan HTTP.
5. Buka terminal atau jendela baris perintah baru pilihan Anda.
6. Untuk membuat kebijakan broker baru, masukkan perintah `curl` baru. Perintah ini mengasumsikan antrean pada vhost `/` default, yang dienkode sebagai `%2F`.

> **Note**
>
> Ganti *`username`* dan *`password`* dengan kredensi masuk administrator broker Anda. Anda mungkin perlu menyesuaikan dan mengkalibrasi nilai `ha-sync-batch-size` (*`100`*) berdasarkan jumlah dan ukuran pesan yang tidak disinkronkan dalam antrian Anda. Mengganti titik akhir broker dengan URL yang Anda catat sebelumnya.

```
curl -i -u username:password -H "content-type:application/json" -XPUT \
-d '{"pattern":".*", "priority":1, "definition":{"ha-sync-batch-size":100, "ha-
mode":"all", "ha-sync-mode":"automatic"}}' \
https://b-589c045f-f8ln-4ab0-a89c-co62e1c32ef8.mq.us-west-2.amazonaws.com/api/
policies/%2F/batch-size-policy
```

7. Untuk mengonfirmasi bahwa kebijakan baru ditambahkan ke kebijakan pengguna broker, masukkan perintah `curl` berikut untuk daftar seluruh kebijakan broker.

```
curl -i -u username:password https://b-589c045f-f8ln-4ab0-a89c-co62e1c32ef8.mq.us-west-2.amazonaws.com/api/policies
```

Langkah 2: Memulai ulang sinkronisasi antrean

Setelah menerapkan kebijakan `ha-sync-batch-size` baru untuk broker Anda, mulai ulang sinkronisasi antrean.

Untuk memulai ulang sinkronisasi antrean menggunakan konsol web RabbitMQ

> **Note**
>
> Untuk membuka konsol web RabbitMQ, lihat petunjuk sebelumnya di Langkah 1 dalam tutorial ini.

1. Di konsol web RabbitMQ, di bagian atas halaman, pilih Antrean.
2. Di halaman Antrean, di bawah Semua antrean, temukan antrean yang dijeda. Di baris Kebijakan, antrian Anda harus mencantumkan nama kebijakan baru yang Anda buat (misalnya,`batch-size-policy`).
3. Untuk memulai ulang proses sinkronisasi dengan ukuran batch yang dikurangi, pertama-tama batalkan sinkronisasi antrian. Kemudian restart sinkronisasi antrian.

> **Note**
>
> Jika sinkronisasi dijeda dan tidak berhasil diselesaikan, coba kurangi nilai `ha-sync-batch-size` dan mulai ulang sinkronisasi antrean lagi.

Langkah selanjutnya

- Setelah antrian Anda berhasil disinkronkan, Anda dapat memantau jumlah memori yang digunakan node RabbitMQ Anda dengan melihat metrik Amazon. CloudWatch `RabbitMQMemUsed` Anda juga dapat melihat metrik `RabbitMQMemLimit` untuk memantau batas memori simpul. Untuk informasi lebih lanjut, lihat Mengakses CloudWatch metrik untuk Amazon MQ dan CloudWatch Metrik yang tersedia untuk Amazon MQ untuk broker RabbitMQ.

- Agar sinkronisasi antrean tidak dijeda, sebaiknya buat antrean tetap pendek dan memproses pesan. Untuk beban kerja dengan ukuran pesan yang lebih besar, kami juga merekomendasikan untuk meningkatkan tipe instans broker ke ukuran instans yang lebih besar dengan lebih banyak memori. Untuk informasi lebih lanjut tentang jenis instans broker dan mengedit preferensi broker, lihatMengedit preferensi broker.
- Ketika Anda membuat broker Amazon MQ for RabbitMQ, Amazon MQ menerapkan serangkaian kebijakan default dan batasan host virtual untuk mengoptimalkan performa broker. Jika broker Anda tidak memiliki kebijakan dan batasan default yang disarankan, sebaiknya buat sendiri. Untuk informasi selengkapnya tentang cara membuat kebijakan default dan batasan vhost, lihat https://docs.aws.amazon.com//amazon-mq/latest/developer-guide/rabbitmq-defaults.html.

Sumber daya terkait

- UpdateBrokerInput— Gunakan properti broker ini untuk memperbarui jenis instans broker menggunakan Amazon MQ API.
- Parameter dan Kebijakan (Dokumentasi Server RabbitMQ) – Pelajari lebih lanjut tentang parameter dan kebijakan RabbitMQ di situs web RabbitMQ.
- HTTP API Manajemen RabbitMQ – Pelajari lebih lanjut tentang API manajemen RabbitMQ.

Mengurangi jumlah koneksi dan saluran

Koneksi ke RabbitMQ Anda di broker Amazon MQ dapat ditutup baik oleh aplikasi klien Anda, atau dengan menutupnya secara manual menggunakan konsol web RabbitMQ. Untuk menutup koneksi menggunakan konsol web RabbitMQ, lakukan hal berikut:

1. Masuk Konsol Manajemen AWS dan buka konsol web RabbitMQ broker Anda.
2. Pada konsol RabbitMQ, pilih tab Connections.
3. Pada halaman Koneksi, di bawah Semua koneksi, pilih nama koneksi yang ingin Anda tutup dari daftar.
4. Pada halaman detail koneksi, pilih Tutup koneksi ini untuk memperluas bagian, lalu pilih Paksa Tutup. Secara opsional, Anda dapat mengganti teks default untuk Alasan dengan deskripsi Anda sendiri. RabbitMQ di Amazon MQ akan mengembalikan alasan yang Anda tentukan ke klien saat Anda menutup koneksi.
5. Pilih OK pada kotak dialog untuk mengonfirmasi dan menutup koneksi.

Saat Anda menutup koneksi, saluran apa pun yang terkait dengan koneksi tertutup juga akan ditutup.

> **Note**
>
> Aplikasi klien Anda dapat dikonfigurasi untuk secara otomatis membangun kembali koneksi ke broker setelah ditutup. Dalam hal ini, menutup koneksi dari konsol web broker tidak akan cukup untuk mengurangi jumlah koneksi atau saluran.

Untuk broker tanpa akses publik, Anda dapat memblokir koneksi sementara dengan menolak lalu lintas masuk pada port protokol pesan yang sesuai, misalnya, port `5671` untuk koneksi AMQP. Anda dapat memblokir port di grup keamanan yang Anda berikan ke Amazon MQ saat membuat broker. Untuk informasi selengkapnya tentang memodifikasi grup keamanan, lihat Menambahkan aturan ke grup keamanan di Panduan Pengguna Amazon VPC.

Langkah 2: Hubungkan aplikasi berbasis JVM ke broker Anda

Setelah membuat broker RabbitMQ, Anda dapat menghubungkan aplikasi ke broker. Contoh berikut menunjukkan cara menggunakan Pustaka klien RabbitMQ Java untuk membuat koneksi ke broker, membuat antrean, dan mengirim pesan. Anda dapat terhubung ke broker RabbitMQ menggunakan pustaka klien RabbitMQ yang didukung untuk berbagai bahasa. Untuk informasi selengkapnya tentang pustaka klien RabbitMQ yang didukung, lihat pustaka klien RabbitMQ dan alat pengembang.

Prasyarat

> **Note**
>
> Langkah-langkah prasyarat berikut ini hanya berlaku untuk broker RabbitMQ yang dibuat tanpa aksesibilitas publik. Jika Anda membuat broker dengan aksesibilitas publik, Anda dapat melewatinya.

Mengaktifkan atribut VPC

Untuk memastikan bahwa broker dapat diakses dalam VPC, Anda harus mengaktifkan atribut VPC `enableDnsHostnames` dan `enableDnsSupport`. Untuk informasi selengkapnya, lihat Dukungan DNS di VPC Anda dalam Panduan Pengguna Amazon VPC.

Mengaktifkan koneksi masuk

1. Masuk ke konsol Amazon MQ.
2. Dari daftar broker, pilih nama broker Anda (misalnya, MyBroker).
3. Pada ***MyBroker***halaman, di bagian Koneksi, catat alamat dan port URL konsol web broker dan protokol tingkat kabel.
4. Di bagian Detail, di bawah Keamanan dan jaringan, pilih nama grup keamanan Anda atau
   
   .

   Halaman Grup Keamanan Dasbor EC2 akan ditampilkan.
5. Dari daftar grup keamanan, pilih grup keamanan Anda.
6. Di bagian bawah halaman, pilih tab Masuk, lalu pilih Edit.
7. Di kotak dialog Edit aturan masuk, tambahkan aturan untuk setiap URL atau titik akhir yang Anda inginkan untuk dapat diakses secara publik (contoh berikut menampilkan cara melakukannya untuk konsol web broker).

   a. Pilih Tambahkan aturan.
   b. Untuk Jenis, pilih TCP Kustom.
   c. Untuk Sumber, biarkan Kustom dipilih lalu ketik alamat IP sistem yang Anda inginkan untuk dapat mengakses konsol web (misalnya, `192.0.2.1`).
   d. Pilih Simpan.

      Broker Anda kini dapat menerima koneksi masuk.

Menambahkan dependensi Java

Jika Anda menggunakan Apache Maven untuk mengotomatisasi build, tambahkan dependensi berikut ke file `pom.xml`. Untuk informasi selengkapnya tentang file Model Objek Proyek di Apache Maven, lihat Pengantar POM.

```
<dependency>
    <groupId>com.rabbitmq</groupId>
    <artifactId>amqp-client</artifactId>
    <version>5.9.0</version>
</dependency>
```

Jika Anda menggunakan Gradle untuk mengotomatisasi build, nyatakan dependensi berikut.

```
dependencies {
    compile 'com.rabbitmq:amqp-client:5.9.0'
}
```

Mengimpor kelas **`Connection`** dan **`Channel`**

Klien RabbitMQ Java menggunakan `com.rabbitmq.client` sebagai paket tingkat atas, dengan kelas API `Connection` dan `Channel`, masing-masing mewakili koneksi dan saluran AMQP 0-9-1. Impor kelas `Connection` dan `Channel` sebelum menggunakannya, seperti yang ditampilkan dalam contoh berikut.

```
import com.rabbitmq.client.Connection;
import com.rabbitmq.client.Channel;
```

Membuat **`ConnectionFactory`** dan menghubungkan ke broker Anda

Gunakan contoh berikut untuk membuat instans kelas `ConnectionFactory` dengan parameter yang diberikan. Gunakan metode `setHost` untuk mengonfigurasi titik akhir broker yang Anda perhatikan sebelumnya. Untuk koneksi tingkat wire `AMQPS`, gunakan port `5671`.

```
ConnectionFactory factory = new ConnectionFactory();

factory.setUsername(username);
factory.setPassword(password);

//Replace the URL with your information
factory.setHost("b-c8352341-ec91-4a78-ad9c-a43f23d325bb.mq.us-west-2.amazonaws.com");
factory.setPort(5671);

// Allows client to establish a connection over TLS
factory.useSslProtocol();

// Create a connection
Connection conn = factory.newConnection();

// Create a channel
Channel channel = conn.createChannel();
```

Memublikasikan pesan ke pertukaran

Anda dapat menggunakan `Channel.basicPublish` untuk memublikasikan pesan ke pertukaran. Contoh berikut menggunakan kelas `Builder` AMQP untuk membangun objek properti pesan dengan jenis konten `plain/text`.

```
byte[] messageBodyBytes = "Hello, world!".getBytes();
channel.basicPublish(exchangeName, routingKey,
             new AMQP.BasicProperties.Builder()
               .contentType("text/plain")
               .userId("userId")
               .build(),
              messageBodyBytes);
```

> **Note**
>
> Perhatikan bahwa `BasicProperties` adalah kelas bagian dalam dari kelas pemilik yang dihasilkan secara otomatis, `AMQP`.

Berlangganan antrian dan menerima pesan

Anda dapat menerima pesan dengan berlangganan antrian menggunakan antarmuka. `Consumer` Setelah berlangganan, pesan kemudian akan dikirim secara otomatis saat mereka tiba.

Cara termudah untuk menerapkan `Consumer` adalah dengan menggunakan subkelas `DefaultConsumer`. Objek `DefaultConsumer` dapat diteruskan sebagai bagian dari panggilan `basicConsume` untuk menyiapkan langganan seperti yang ditampilkan dalam contoh berikut.

```
boolean autoAck = false;
channel.basicConsume(queueName, autoAck, "myConsumerTag",
     new DefaultConsumer(channel) {
         @Override
         public void handleDelivery(String consumerTag,
                                    Envelope envelope,
                                    AMQP.BasicProperties properties,
                                    byte[] body)
             throws IOException
         {
             String routingKey = envelope.getRoutingKey();
             String contentType = properties.getContentType();
```

```
            long deliveryTag = envelope.getDeliveryTag();
            // (process the message components here ...)
            channel.basicAck(deliveryTag, false);
        }
    });
```

> **Note**
>
> Karena kita menentukan `autoAck = false`, pesan yang dikirim ke `Consumer` perlu diakui, dan paling mudah dilakukan dengan metode `handleDelivery`, seperti yang ditampilkan dalam contoh.

Menutup koneksi Anda dan memutuskan koneksi dari broker

Untuk memutuskan koneksi dari broker RabbitMQ, tutup saluran dan koneksi seperti yang ditunjukkan di bawah ini.

```
channel.close();
conn.close();
```

> **Note**
>
> Untuk informasi selengkapnya tentang bekerja dengan pustaka klien Java RabbitMQ, lihat Panduan API Klien Java RabbitMQ.

Langkah 3: (Opsional) Connect ke AWS Lambda fungsi

AWS Lambda dapat terhubung ke dan mengkonsumsi pesan dari broker Amazon MQ Anda. Saat Anda menghubungkan broker ke Lambda, Anda membuat pemetaan sumber peristiwa yang membaca pesan dari antrian dan memanggil fungsi secara sinkron. Pemetaan sumber acara yang Anda buat membaca pesan dari broker Anda dalam batch dan mengubahnya menjadi muatan Lambda dalam bentuk objek JSON.

Untuk menghubungkan broker Anda ke fungsi Lambda

1. Tambahkan izin peran IAM berikut ke peran eksekusi fungsi Lambda Anda.
   - mq: DescribeBroker

- EC2: CreateNetworkInterface
- EC2: DeleteNetworkInterface
- EC2: DescribeNetworkInterfaces
- EC2: DescribeSecurityGroups
- EC2: DescribeSubnets
- EC2: DescribeVpcs
- log: CreateLogGroup
- log: CreateLogStream
- log: PutLogEvents
- manajer rahasia: GetSecretValue

> **Note**
>
> Tanpa izin IAM yang diperlukan, fungsi Anda tidak akan berhasil membaca catatan dari sumber daya Amazon MQ.

2. (Opsional) Jika Anda telah membuat broker tanpa aksesibilitas publik, Anda harus melakukan salah satu hal berikut untuk memungkinkan Lambda terhubung ke broker Anda:

   - Konfigurasikan satu NAT gateway per subnet publik. Untuk informasi selengkapnya, lihat Akses Internet dan layanan untuk fungsi yang terhubung dengan VPC di Panduan Pengembang.AWS Lambda
   - Buat koneksi antara Amazon Virtual Private Cloud (Amazon VPC) dan Lambda menggunakan titik akhir VPC. VPC Amazon Anda juga harus terhubung ke AWS Security Token Service (AWS STS) dan titik akhir Secrets Manager. Untuk informasi selengkapnya, lihat Mengonfigurasi titik akhir VPC antarmuka untukAWS Lambda Lambda di Panduan Pengembang.

3. Konfigurasikan broker Anda sebagai sumber acara untuk fungsi Lambda menggunakan. Konsol Manajemen AWS Anda juga dapat menggunakan `create-event-source-mapping` AWS Command Line Interface perintah.

4. Tulis beberapa kode untuk fungsi Lambda Anda untuk memproses pesan dari yang Anda konsumsi dari broker Anda. Payload Lambda yang diambil oleh pemetaan sumber acara Anda tergantung pada jenis mesin broker. Berikut ini adalah contoh payload Lambda untuk Amazon MQ untuk antrian RabbitMQ.

> **Note**
>
> Dalam contoh, `test` adalah nama antrian, dan `/` merupakan nama host virtual default. Saat menerima pesan, sumber acara mencantumkan pesan di bawah`test::/`.


```
{
  "eventSource": "aws:rmq",
  "eventSourceArn": "arn:aws:mq:us-
west-2:112556298976:broker:test:b-9bcfa592-423a-4942-879d-eb284b418fc8",
  "rmqMessagesByQueue": {
    "test::/": [
      {
        "basicProperties": {
          "contentType": "text/plain",
          "contentEncoding": null,
          "headers": {
            "header1": {
              "bytes": [
                118,
                97,
                108,
                117,
                101,
                49
              ]
            },
            "header2": {
              "bytes": [
                118,
                97,
                108,
                117,
                101,
                50
              ]
            },
            "numberInHeader": 10
          }
          "deliveryMode": 1,
          "priority": 34,
```

```
          "correlationId": null,
          "replyTo": null,
          "expiration": "60000",
          "messageId": null,
          "timestamp": "Jan 1, 1970, 12:33:41 AM",
          "type": null,
          "userId": "AIDACKCEVSQ6C2EXAMPLE",
          "appId": null,
          "clusterId": null,
          "bodySize": 80
        },
        "redelivered": false,
        "data": "eyJ0aW1lb3V0IjowLCJkYXRhIjoiQ1pybWYwR3c4T3Y0YnFMUXhENEUifQ=="
      }
    ]
  }
}
```


Untuk informasi selengkapnya tentang menghubungkan Amazon MQ ke Lambda, opsi yang didukung Lambda untuk sumber peristiwa Amazon MQ, dan kesalahan pemetaan sumber peristiwa, lihat Menggunakan Lambda dengan Amazon MQ di Panduan Pengembang.AWS Lambda

Menggunakan otentikasi OAuth 2.0 dan otorisasi untuk Amazon MQ untuk RabbitMQ

Tutorial ini menjelaskan cara mengonfigurasi otentikasi OAuth 2.0 untuk Amazon MQ Anda untuk broker RabbitMQ menggunakan Amazon Cognito sebagai penyedia 2.0. OAuth

> **Note**
>
> Amazon Cognito tidak tersedia di Tiongkok (Beijing) dan Tiongkok (Ningxia).

> **Important**
>
> Tutorial ini khusus untuk Amazon Cognito, tetapi Anda dapat menggunakan penyedia identitas lain ()IdPs. Untuk informasi selengkapnya, lihat Contoh Otentikasi OAuth 2.0.

Di halaman ini

- Prasyarat untuk mengonfigurasi otentikasi 2.0 OAuth
- Mengonfigurasi otentikasi OAuth 2.0 dengan Amazon Cognito menggunakan AWS CLI
- Mengkonfigurasi OAuth 2.0 dan otentikasi sederhana dengan Amazon Cognito

Prasyarat untuk mengonfigurasi otentikasi 2.0 OAuth

Anda dapat mengatur sumber daya Amazon Cognito yang diperlukan dalam tutorial ini dengan menerapkan tumpukan, AWS CDK tumpukan Amazon Cognito untuk plugin RabbitMQ 2. OAuth Jika Anda menyiapkan Amazon Cognito secara manual, pastikan Anda memenuhi prasyarat berikut sebelum mengonfigurasi 2.0 OAuth di Amazon MQ Anda untuk broker RabbitMQ:

Prasyarat untuk mengatur Amazon Cognito

- Siapkan titik akhir Amazon Cognito dengan membuat kumpulan pengguna. Untuk melakukan ini, lihat blog berjudul Cara menggunakan OAuth 2.0 di Amazon Cognito: Pelajari tentang hibah 2.0 yang OAuth berbeda.
- Buat server sumber daya yang dipanggil `rabbitmq` di kumpulan pengguna dengan cakupan berikut yang ditentukan:`read:all`,, `write:allconfigure:all`, dan`tag:administrator`. Cakupan ini akan dikaitkan dengan izin RabbitMQ.

  Untuk informasi tentang membuat server sumber daya, lihat Mendefinisikan server sumber daya untuk kumpulan pengguna (Konsol Manajemen AWS) di Panduan Pengembang Amazon Cognito.
- Buat klien aplikasi berikut:
  - Klien aplikasi untuk jenis kumpulan penggunaMachine-to-Machine application. Ini adalah klien rahasia dengan rahasia klien yang akan digunakan untuk klien RabbitMQ AMQP. Untuk informasi selengkapnya tentang klien aplikasi dan membuatnya, lihat Jenis klien aplikasi dan Membuat klien aplikasi.
  - Klien aplikasi untuk jenis kumpulan penggunaSingle-page application. Ini adalah klien publik yang akan digunakan untuk masuk pengguna ke konsol manajemen RabbitMQ. Anda harus memperbarui klien aplikasi ini untuk menyertakan titik akhir Amazon MQ untuk broker RabbitMQ yang akan Anda buat dalam prosedur berikut sebagai URL panggilan balik yang diizinkan. Untuk informasi selengkapnya, lihat Menyiapkan login terkelola dengan konsol Amazon Cognito.

Prasyarat untuk mengatur Amazon MQ

- Instalasi Docker yang berfungsi untuk menjalankan skrip bash yang memverifikasi apakah pengaturan OAuth 2.0 berhasil atau tidak.
- AWS CLI versi >= `2.28.23` untuk membuat penambahan nama pengguna dan kata sandi opsional selama pembuatan broker.

Mengonfigurasi otentikasi OAuth 2.0 dengan Amazon Cognito menggunakan AWS CLI

Prosedur berikut menunjukkan cara mengatur otentikasi OAuth 2.0 untuk Amazon MQ Anda untuk broker RabbitMQ menggunakan Amazon Cognito sebagai IDP. Prosedur ini digunakan AWS CLI untuk membuat dan mengkonfigurasi sumber daya yang diperlukan.

Dalam prosedur berikut, pastikan untuk mengganti nilai placeholder, seperti ConfigurationId dan Revision, *<c-fa3390a5-7e01-4559-ae0c-eb15b38b22ca>* dan*<2>*, dengan nilai sebenarnya.

1. Buat konfigurasi baru menggunakan AWS CLI perintah create-configuration seperti yang ditunjukkan pada contoh berikut.

```
aws mq create-configuration \
  --name "rabbitmq-oauth2-config" \
  --engine-type "RABBITMQ" \
  --engine-version "3.13"
```

Perintah ini mengembalikan respon mirip dengan contoh berikut.

```
{
    "Arn": "arn:aws:mq:us-west-2:123456789012:configuration:c-fa3390a5-7e01-4559-
ae0c-eb15b38b22ca",
    "AuthenticationStrategy": "simple",
    "Created": "2025-07-17T16:03:01.759943+00:00",
    "Id": "c-fa3390a5-7e01-4559-ae0c-eb15b38b22ca",
    "LatestRevision": {
    "Created": "2025-07-17T16:03:01.759000+00:00",
    "Description": "Auto-generated default for rabbitmq-oauth2-config on RabbitMQ
 3.13",
    "Revision": 1
    },
    "Name": "rabbitmq-oauth2-config"
}
```

2. Buat file konfigurasi yang dipanggil **`rabbitmq.conf`** untuk menggunakan OAuth 2.0 sebagai metode otentikasi dan otorisasi, seperti yang ditunjukkan pada contoh berikut.

```
auth_backends.1 = oauth2

# FIXME: Update this value with the token signing key URL of your Amazon Cognito
 user pool.
# If you used the AWS CDK stack to deploy Amazon Cognito, this is one of the stack
 outputs.
auth_oauth2.jwks_url = ${RabbitMqOAuth2TestStack.JwksUri}
auth_oauth2.resource_server_id = rabbitmq
# Amazon Cognito does not include an audience field in access tokens
auth_oauth2.verify_aud = false

# Amazon Cognito does not allow * in its custom scopes. Use aliases to translate
 between Amazon Cognito and RabbitMQ.
auth_oauth2.scope_prefix = rabbitmq/
auth_oauth2.scope_aliases.1.alias = rabbitmq/read:all
auth_oauth2.scope_aliases.1.scope = rabbitmq/read:*/*
auth_oauth2.scope_aliases.2.alias = rabbitmq/write:all
auth_oauth2.scope_aliases.2.scope = rabbitmq/write:*/*
auth_oauth2.scope_aliases.3.alias = rabbitmq/configure:all
auth_oauth2.scope_aliases.3.scope = rabbitmq/configure:*/*

# Allow OAuth 2.0 login for RabbitMQ management console
management.oauth_enabled = true
# FIXME: Update this value with the client ID of your public application client
management.oauth_client_id
 = ${RabbitMqOAuth2TestStack.ManagementConsoleAppClientId}
# FIXME: Update this value with the base JWKS URI (without /.well-known/jwks.json)
auth_oauth2.issuer = ${RabbitMqOAuth2TestStack.Issuer}
management.oauth_scopes = rabbitmq/tag:administrator
```

Konfigurasi ini menggunakan alias cakupan untuk memetakan cakupan yang ditentukan di Amazon Cognito ke cakupan yang kompatibel dengan RabbitMQ.

3. Perbarui konfigurasi menggunakan AWS CLI perintah update-configuration seperti yang ditunjukkan pada contoh berikut. Dalam perintah ini, tambahkan ID konfigurasi yang Anda terima sebagai respons Langkah 1 dari prosedur ini. Misalnya, **`c-fa3390a5-7e01-4559-ae0c-eb15b38b22ca`**.

```
aws mq update-configuration \
```

```
  --configuration-id "<c-fa3390a5-7e01-4559-ae0c-eb15b38b22ca>" \
  --data "$(cat rabbitmq.conf | base64 --wrap=0)"
```

Perintah ini mengembalikan respon mirip dengan contoh berikut.

```
{
    "Arn": "arn:aws:mq:us-west-2:123456789012:configuration:c-b600ac8e-8183-4f74-
a713-983e59f30e3d",
    "Created": "2025-07-17T16:57:04.520931+00:00",
    "Id": "c-b600ac8e-8183-4f74-a713-983e59f30e3d",
    "LatestRevision": {
        "Created": "2025-07-17T16:57:39.172000+00:00",
        "Revision": 2
    },
    "Name": "rabbitmq-oauth2-config",
    "Warnings": []
}
```

4. Buat broker dengan konfigurasi OAuth 2.0 yang Anda buat di Langkah 2 prosedur ini. Untuk melakukan ini, gunakan AWS CLI perintah create-broker seperti yang ditunjukkan pada contoh berikut. Dalam perintah ini, berikan ID konfigurasi dan nomor revisi yang Anda peroleh dalam tanggapan Langkah 1 dan 2 masing-masing. Misalnya, **`c-fa3390a5-7e01-4559-ae0c-eb15b38b22ca`** dan **2**.

```
aws mq create-broker \
 --broker-name "rabbitmq-oauth2-broker" \
 --engine-type "RABBITMQ" \
 --engine-version "3.13" \
 --host-instance-type "mq.m7g.large" \
 --deployment-mode "CLUSTER_MULTI_AZ" \
 --logs '{"General": true}' \
 --publicly-accessible \
 --configuration '{"Id": "<c-fa3390a5-7e01-4559-ae0c-
eb15b38b22ca>","Revision": <2>}' \
```

Perintah ini mengembalikan respon mirip dengan contoh berikut.

```
{
    "BrokerArn": "arn:aws:mq:us-west-2:123456789012:broker:rabbitmq-oauth2-
broker:b-2a1b5133-a10c-49d2-879b-8c176c34cf73",
    "BrokerId": "b-2a1b5133-a10c-49d2-879b-8c176c34cf73"
```

```
}
```

5. Verifikasi bahwa status broker bertransisi dari `CREATION_IN_PROGRESS` ke`RUNNING`, menggunakan AWS CLI perintah deskripsi-broker seperti yang ditunjukkan pada contoh berikut. Dalam perintah ini, berikan ID broker yang Anda peroleh dalam hasil langkah sebelumnya Misalnya,**`b-2a1b5133-a10c-49d2-879b-8c176c34cf73`**.

```
aws mq describe-broker \
 --broker-id "<b-2a1b5133-a10c-49d2-879b-8c176c34cf73>"
```

Perintah ini mengembalikan respon mirip dengan contoh berikut. Respons berikut adalah versi singkat dari output lengkap yang dikembalikan `describe-broker` perintah. Tanggapan ini menunjukkan status broker dan strategi otentikasi yang digunakan untuk mengamankan broker. Dalam hal ini, strategi `config_managed` otentikasi menunjukkan bahwa broker menggunakan OAuth 2 metode otentikasi.

```
{
    "AuthenticationStrategy": "config_managed",
    ...,
    "BrokerState": "RUNNING",
    ...
}
```

Untuk masuk ke konsol manajemen RabbitMQ menggunakan OAuth2, titik akhir broker perlu ditambahkan sebagai URL panggilan balik yang valid di klien aplikasi Amazon Cognito yang sesuai. Untuk informasi selengkapnya, lihat Langkah 5 dalam pengaturan sampel tumpukan CDK Amazon Cognito kami.

6. Verifikasi otentikasi dan otorisasi OAuth 2.0 dengan skrip berikut`perf-test.sh`.

Gunakan skrip bash ini untuk menguji konektivitas ke Amazon MQ Anda untuk broker RabbitMQ. Skrip ini memperoleh token dari Amazon Cognito dan memverifikasi apakah koneksi telah dikonfigurasi dengan benar. Jika berhasil dikonfigurasi, Anda akan melihat broker Anda mempublikasikan dan menggunakan pesan.

Jika Anda menerima `ACCESS_REFUSED` kesalahan, Anda dapat memecahkan masalah pengaturan konfigurasi Anda dengan menggunakan CloudWatch log untuk broker Anda. Anda dapat menemukan tautan untuk grup CloudWatch log untuk broker Anda di konsol Amazon MQ.

Dalam skrip ini, Anda harus memberikan nilai-nilai berikut:

- `CLIENT_ID`dan`CLIENT_SECRET`: Anda dapat menemukan nilai-nilai ini di halaman Klien aplikasi di konsol Amazon Cognito.
- Domain Cognito: Anda dapat menemukannya di konsol Amazon Cognito. Di bawah Branding, pilih Domain. Pada halaman Domain, Anda dapat menemukan nilai ini di bawah bagian Server sumber daya.
- Titik akhir broker Amazon MQ: Anda dapat menemukan nilai ini di bawah Koneksi di halaman detail broker konsol Amazon MQ.

```
#! /bin/bash
set -e

# Client information
## FIXME: Update this value with the client ID and secret of your confidential
 application client
CLIENT_ID=${RabbitMqOAuth2TestStack.AmqpAppClientId}
CLIENT_SECRET=${RabbitMqOAuth2TestStack.AmqpAppClientSecret}

# FIXME: Update this value with the domain of your Amazon Cognito user pool
RESPONSE=$(curl -X POST ${RabbitMqOAuth2TestStack.TokenEndpoint} \
                -H "Content-Type: application/x-www-form-urlencoded" \
                -d
 "grant_type=client_credentials&client_id=${CLIENT_ID}&client_secret=${CLIENT_SECRET}&scope
configure:all rabbitmq/read:all rabbitmq/tag:administrator rabbitmq/write:all")


# Extract the access_token from the response.
# This token will be passed in the password field when connecting to the broker.
# Note that the username is left blank, the field is ignored by the plugin.
BROKER_PASSWORD=$(echo ${RESPONSE} | jq -r '.access_token')

# FIXME: Update this value with the endpoint of your broker. For
 example, b-89424106-7e0e-4abe-8e98-8de0dada7630.mq.us-east-1.on.aws.
BROKER_DNS=<broker_dns>
CONNECTION_STRING=amqps://:${BROKER_PASSWORD}@${BROKER_DNS}:5671

# Produce/consume messages using the above connection string
QUEUES_COUNT=1
PRODUCERS_COUNT=1
CONSUMERS_COUNT=1
PRODUCER_RATE=1
```

```
docker run -it --rm --ulimit nofile=40960:40960 pivotalrabbitmq/perf-test:latest \
    --queue-pattern 'test-queue-%d' --queue-pattern-from 1 --queue-pattern-to
 $QUEUES_COUNT \
    --producers $PRODUCERS_COUNT --consumers $CONSUMERS_COUNT \
    --id "test${QUEUES_COUNT}q${PRODUCERS_COUNT}p${CONSUMERS_COUNT}c
${PRODUCER_RATE}r" \
    --uri ${CONNECTION_STRING} \
    --flag persistent --rate $PRODUCER_RATE
```

Mengkonfigurasi OAuth 2.0 dan otentikasi sederhana dengan Amazon Cognito

Saat Anda membuat broker dengan otentikasi OAuth 2.0, Anda dapat menentukan salah satu metode otentikasi berikut:

- OAuth 2.0 saja: Untuk menggunakan metode ini, jangan berikan nama pengguna dan kata sandi saat membuat broker. Prosedur sebelumnya menunjukkan cara menggunakan hanya metode otentikasi OAuth 2.0.
- Baik OAuth 2.0 dan otentikasi sederhana: Untuk menggunakan metode ini, berikan nama pengguna dan kata sandi saat membuat broker. Juga, tambahkan `auth_backends.2 = internal` ke konfigurasi broker Anda, seperti yang ditunjukkan dalam prosedur berikut.

Dalam prosedur berikut, pastikan untuk mengganti nilai placeholder, seperti *<ConfigurationId>* dan*<Revision>*, dengan nilai aktualnya.

1. Untuk menggunakan kedua metode otentikasi, buat konfigurasi broker Anda, seperti yang ditunjukkan pada contoh berikut.

```
auth_backends.1 = oauth2
auth_backends.2 = internal

# FIXME: Update this value with the token signing key URL of your Amazon Cognito
 user pool
auth_oauth2.jwks_url = ${RabbitMqOAuth2TestStack.JwksUri}
auth_oauth2.resource_server_id = rabbitmq
auth_oauth2.verify_aud = false

auth_oauth2.scope_prefix = rabbitmq/
auth_oauth2.scope_aliases.1.alias = rabbitmq/read:all
auth_oauth2.scope_aliases.1.scope = rabbitmq/read:*/*
```

```
auth_oauth2.scope_aliases.2.alias = rabbitmq/write:all
auth_oauth2.scope_aliases.2.scope = rabbitmq/write:*/*
auth_oauth2.scope_aliases.3.alias = rabbitmq/configure:all
auth_oauth2.scope_aliases.3.scope = rabbitmq/configure:*/*
```

Konfigurasi ini menggunakan alias cakupan untuk memetakan cakupan yang ditentukan di Amazon Cognito ke cakupan yang kompatibel dengan RabbitMQ.

2. Buat broker yang menggunakan kedua metode otentikasi, seperti yang ditunjukkan pada contoh berikut.

```
aws mq create-broker \
 --broker-name "rabbitmq-oauth2-broker-with-internal-user" \
 --engine-type "RABBITMQ" \
 --engine-version "3.13" \
 --host-instance-type "mq.m7g.large" \
 --deployment-mode "CLUSTER_MULTI_AZ" \
 --logs '{"General": true}' \
 --publicly-accessible \
 --configuration '{"Id": "<ConfigurationId>","Revision": <Revision>}' \
 --users '[{"Username":"<myUser>","Password":"<myPassword11>"}]'
```

3. Verifikasi status broker dan konfigurasi untuk menyiapkan metode otentikasi berhasil seperti yang dijelaskan dalam Langkah 5 dan 6 dari Mengonfigurasi otentikasi OAuth 2.0 dengan Amazon Cognito prosedur.

Menggunakan otentikasi dan otorisasi IAM untuk Amazon MQ untuk RabbitMQ

Prosedur berikut menunjukkan cara mengaktifkan otentikasi dan otorisasi AWS IAM untuk Amazon MQ untuk broker RabbitMQ. Setelah mengaktifkan IAM, pengguna dapat mengautentikasi menggunakan kredensional AWS IAM untuk mengakses API Manajemen RabbitMQ dan terhubung melalui AMQP. Untuk detail tentang cara kerja otentikasi IAM dengan Amazon MQ untuk RabbitMQ, lihat. the section called “Otentikasi dan otorisasi IAM”

Prasyarat

- AWS kredensi administrator untuk AWS akun yang memiliki Amazon MQ untuk broker RabbitMQ
- Lingkungan shell yang dikonfigurasi dengan kredensi administrator ini (menggunakan profil AWS CLI atau variabel lingkungan)

- AWS CLI diinstal dan dikonfigurasi
- `jq`prosesor JSON baris perintah diinstal
- `curl`alat baris perintah diinstal

Mengkonfigurasi otentikasi dan otorisasi IAM menggunakan AWS CLI

1. Tetapkan variabel lingkungan

   Tetapkan variabel lingkungan yang diperlukan untuk broker Anda:

   ```
   export AWS_DEFAULT_REGION=<region>
   export BROKER_ID=<broker-id>
   ```

2. Aktifkan token JWT keluar

   Aktifkan federasi identitas web keluar untuk AWS akun Anda:

   ```
   ISSUER_IDENTIFIER=$(aws iam enable-outbound-web-identity-federation --query
    'IssuerIdentifier' --output text)
   echo $ISSUER_IDENTIFIER
   ```

   Output menampilkan URL pengenal penerbit unik untuk akun Anda dalam format. `https://<id>.tokens.sts.global.api.aws`

3. Buat dokumen kebijakan IAM

   Buat dokumen kebijakan yang memberikan izin untuk mendapatkan token identitas web:

   ```
   cat > policy.json << 'EOF'
   {
       "Version": "2012-10-17",
       "Statement": [
           {
               "Sid": "VisualEditor0",
               "Effect": "Allow",
               "Action": [
   ```

```
                "sts:GetWebIdentityToken",
                "sts:TagGetWebIdentityToken"
            ],
            "Resource": "*"
        }
    ]
}
EOF
```

4. Buat kebijakan kepercayaan

   Ambil identitas penelepon Anda dan buat dokumen kebijakan kepercayaan:

```
CALLER_ARN=$(aws sts get-caller-identity --query Arn --output text)
cat > trust-policy.json << EOF
{
    "Version": "2012-10-17",
    "Statement": [
        {
            "Effect": "Allow",
            "Principal": {
                "AWS": "$CALLER_ARN"
            },
            "Action": "sts:AssumeRole"
        }
    ]
}
EOF
```

5. Buat peran IAM

   Buat peran IAM dan lampirkan kebijakan:

```
aws iam create-role --role-name RabbitMqAdminRole --assume-role-policy-document
 file://trust-policy.json
aws iam put-role-policy --role-name RabbitMqAdminRole --policy-name
 RabbitMqAdminRolePolicy --policy-document file://policy.json
```

6. Konfigurasikan pengaturan OAuth2 RabbitMQ

   Buat file konfigurasi RabbitMQ dengan pengaturan OAuth2 otentikasi dan otorisasi:

```
cat > rabbitmq.conf << EOF
auth_backends.1 = oauth2
auth_backends.2 = internal

auth_oauth2.jwks_url = ${ISSUER_IDENTIFIER}/.well-known/jwks.json
auth_oauth2.resource_server_id = rabbitmq
auth_oauth2.scope_prefix = rabbitmq/

auth_oauth2.additional_scopes_key = sub
auth_oauth2.scope_aliases.1.alias = arn:aws:iam::$(aws sts get-caller-identity --
query Account --output text):role/RabbitMqAdminRole
auth_oauth2.scope_aliases.1.scope = rabbitmq/tag:administrator rabbitmq/read:*/*
 rabbitmq/write:*/* rabbitmq/configure:*/*
auth_oauth2.https.hostname_verification = wildcard

management.oauth_enabled = true
EOF
```

7. Perbarui konfigurasi broker

   Terapkan konfigurasi baru ke broker Anda:

```
# Retrieve the configuration ID
CONFIG_ID=$(aws mq describe-broker --broker-id $BROKER_ID --query
 'Configurations[0].Id' --output text)

# Create a new configuration revision
REVISION=$(aws mq update-configuration --configuration-id $CONFIG_ID --data "$(cat
 rabbitmq.conf | base64 --wrap=0)" --query 'LatestRevision.Revision' --output text)

# Apply the configuration to the broker
aws mq update-broker --broker-id $BROKER_ID --configuration Id=$CONFIG_ID,Revision=
$REVISION

# Reboot the broker to apply changes
aws mq reboot-broker --broker-id $BROKER_ID
```

Tunggu status broker kembali `RUNNING` sebelum melanjutkan ke langkah berikutnya.

8. Dapatkan token JWT

   Asumsikan peran IAM dan dapatkan token identitas web:

```
# Assume the RabbitMqAdminRole
ROLE_CREDS=$(aws sts assume-role --role-arn arn:aws:iam::$(aws sts get-caller-
identity --query Account --output text):role/RabbitMqAdminRole --role-session-name
 rabbitmq-session)

# Configure the session with temporary credentials
export AWS_ACCESS_KEY_ID=$(echo "$ROLE_CREDS" | jq -r '.Credentials.AccessKeyId')
export AWS_SECRET_ACCESS_KEY=$(echo "$ROLE_CREDS" | jq -r
 '.Credentials.SecretAccessKey')
export AWS_SESSION_TOKEN=$(echo "$ROLE_CREDS" | jq -r '.Credentials.SessionToken')

# Obtain the web identity token
TOKEN_RESPONSE=$(aws sts get-web-identity-token \
    --audience "rabbitmq" \
    --signing-algorithm ES384 \
    --duration-seconds 300 \
    --tags Key=scope,Value="rabbitmq/tag:administrator")

# Extract the token
TOKEN=$(echo "$TOKEN_RESPONSE" | jq -r '.WebIdentityToken')
```

9. Akses API Manajemen RabbitMQ

   Gunakan token JWT untuk mengakses API Manajemen RabbitMQ:

```
BROKER_URL=<broker-id>.mq.<region>.on.aws

curl -u ":$TOKEN" \
    -X GET https://${BROKER_URL}/api/overview \
    -H "Content-Type: application/json"
```

Respons yang berhasil mengonfirmasi bahwa otentikasi IAM berfungsi dengan benar. Tanggapan tersebut berisi informasi ikhtisar broker dalam format JSON.

10. Connect melalui AMQP menggunakan token JWT

    Uji konektivitas AMQP menggunakan token JWT dengan alat uji sempurna:

```
BROKER_DNS=<broker-endpoint>
CONNECTION_STRING=amqps://:${TOKEN}@${BROKER_DNS}:5671

docker run -it --rm --ulimit nofile=40960:40960 pivotalrabbitmq/perf-test:latest \
    --queue-pattern 'test-queue-%d' --queue-pattern-from 1 --queue-pattern-to 1 \
    --producers 1 --consumers 1 \
    --uri ${CONNECTION_STRING} \
    --flag persistent --rate 1
```

    Jika Anda menerima `ACCESS_REFUSED` kesalahan, Anda dapat memecahkan masalah pengaturan konfigurasi Anda dengan menggunakan CloudWatch log untuk broker Anda. Anda dapat menemukan tautan untuk grup CloudWatch log Log untuk broker Anda di konsol Amazon MQ.

Menggunakan otentikasi dan otorisasi LDAP untuk Amazon MQ untuk RabbitMQ

Tutorial ini menjelaskan cara mengkonfigurasi otentikasi dan otorisasi LDAP untuk Amazon MQ Anda untuk broker RabbitMQ menggunakan. AWS Managed Microsoft AD

Di halaman ini

- Prasyarat untuk mengonfigurasi otentikasi dan otorisasi LDAP
- Mengkonfigurasi LDAP di RabbitMQ menggunakan CLI AWS

Prasyarat untuk mengonfigurasi otentikasi dan otorisasi LDAP

Anda dapat mengatur AWS sumber daya yang diperlukan dalam tutorial ini dengan menerapkan tumpukan AWS CDK untuk Amazon MQ untuk integrasi RabbitMQ LDAP dengan. AWS Managed Microsoft AD

Tumpukan CDK ini secara otomatis membuat semua AWS sumber daya yang diperlukan termasuk AWS Managed Microsoft AD, pengguna dan grup LDAP, Network Load Balancer, sertifikat, dan peran IAM. Lihat paket README untuk daftar lengkap sumber daya yang dibuat oleh tumpukan.

Jika Anda menyiapkan sumber daya secara manual alih-alih menggunakan tumpukan CDK, pastikan Anda memiliki infrastruktur yang setara sebelum mengonfigurasi LDAP di Amazon MQ Anda untuk broker RabbitMQ.

Prasyarat untuk mengatur Amazon MQ

AWS Versi CLI >= 2.28.23 untuk membuat penambahan nama pengguna dan kata sandi opsional selama pembuatan broker.

Mengkonfigurasi LDAP di RabbitMQ menggunakan CLI AWS

Prosedur ini menggunakan AWS CLI untuk membuat dan mengkonfigurasi sumber daya yang diperlukan. Dalam prosedur berikut, pastikan untuk mengganti nilai placeholder, seperti ConfigurationId dan Revision, <c-fa3390a5-7e01-4559-ae0c-eb15b38b22ca> dan<2>, dengan nilai sebenarnya.

1. Buat konfigurasi baru menggunakan perintah `create-configuration` AWS CLI seperti yang ditunjukkan pada contoh berikut.

```
aws mq create-configuration \
  --name "rabbitmq-ldap-config" \
  --engine-type "RABBITMQ" \
  --engine-version "3.13"
```

   Perintah ini mengembalikan respon mirip dengan contoh berikut.

```
{
"Arn": "arn:aws:mq:us-west-2:123456789012:configuration:c-fa3390a5-7e01-4559-ae0c-
eb15b38b22ca",
    "AuthenticationStrategy": "simple",
    "Created": "2025-07-17T16:03:01.759943+00:00",
    "Id": "c-fa3390a5-7e01-4559-ae0c-eb15b38b22ca",
    "LatestRevision": {
"Created": "2025-07-17T16:03:01.759000+00:00",
```

```
    "Description": "Auto-generated default for rabbitmq-ldap-config on RabbitMQ
 3.13",
    "Revision": 1
    },
    "Name": "rabbitmq-ldap-config"
}
```

2. Buat file konfigurasi yang dipanggil `rabbitmq.conf` untuk menggunakan LDAP sebagai metode otentikasi dan otorisasi, seperti yang ditunjukkan pada contoh berikut. Ganti semua nilai placeholder dalam template (ditandai dengan`${RabbitMqLdapTestStack.*}`) dengan nilai aktual dari output tumpukan AWS CDK prasyarat yang diterapkan atau infrastruktur yang setara.

```
auth_backends.1 = ldap

# LDAP authentication settings - For more information,
# see https://www.rabbitmq.com/docs/ldap#basic

# FIXME: Replace the ${RabbitMqLdapTestStack.*} placeholders with actual values
# from your deployed prerequisite CDK stack outputs.
auth_ldap.servers.1 = ${RabbitMqLdapTestStack.NlbDnsName}
auth_ldap.dn_lookup_bind.user_dn = ${RabbitMqLdapTestStack.DnLookupUserDn}
auth_ldap.dn_lookup_base = ${RabbitMqLdapTestStack.DnLookupBase}
auth_ldap.dn_lookup_attribute = ${RabbitMqLdapTestStack.DnLookupAttribute}
auth_ldap.port = 636
auth_ldap.use_ssl = true
auth_ldap.ssl_options.verify = verify_peer
auth_ldap.log = network

# AWS integration for secure credential retrieval
# - see: https://github.com/amazon-mq/rabbitmq-aws
# The aws plugin allows RabbitMQ to securely retrieve credentials and certificates
# from AWS services.

# Replace the ${RabbitMqLdapTestStack.*} placeholders with actual ARN values
# from your deployed prerequisite CDK stack outputs.
aws.arns.auth_ldap.ssl_options.cacertfile = ${RabbitMqLdapTestStack.CaCertArn}
aws.arns.auth_ldap.dn_lookup_bind.password =
 ${RabbitMqLdapTestStack.DnLookupUserPasswordArn}
aws.arns.assume_role_arn = ${RabbitMqLdapTestStack.AmazonMqAssumeRoleArn}

# LDAP authorization queries - For more information,
```

```
# see: https://www.rabbitmq.com/docs/ldap#authorisation

# FIXME: Replace the ${RabbitMqLdapTestStack.*} placeholders with actual group DN
# values from your deployed prerequisite CDK stack outputs
# Uses Active Directory groups created by the prerequisite CDK stack
auth_ldap.queries.tags = '''
[{administrator, {in_group,
 "${RabbitMqLdapTestStack.RabbitMqAdministratorsGroupDn}"}},
{management,    {in_group,
 "${RabbitMqLdapTestStack.RabbitMqMonitoringUsersGroupDn}"}}]
'''

# FIXME: This provides all authenticated users access to all vhosts
# - update to restrict access as required
auth_ldap.queries.vhost_access = '''
{constant, true}
'''

# FIXME: This provides all authenticated users full access to all
# queues and exchanges - update to restrict access as required
auth_ldap.queries.resource_access = '''
{for, [    {permission, configure, {constant, true}},
     {permission, write,
      {for, [{resource, queue,    {constant, true}},
             {resource, exchange, {constant, true}}]}},
     {permission, read,
      {for, [{resource, exchange, {constant, true}},
             {resource, queue,    {constant, true}}]}}
    ]
}
'''

# FIXME: This provides all authenticated users access to all topics
# - update to restrict access as required
auth_ldap.queries.topic_access = '''
{for, [{permission, write, {constant, true}},
     {permission, read,  {constant, true}}
    ]
}
'''
```

3. Perbarui konfigurasi menggunakan perintah `update-configuration` AWS CLI seperti yang ditunjukkan pada contoh berikut. Dalam perintah ini, tambahkan ID konfigurasi yang Anda terima

sebagai respons Langkah 1 dari prosedur ini. Misalnya, `c-fa3390a5-7e01-4559-ae0c-eb15b38b22ca`.

```
aws mq update-configuration \
  --configuration-id "<c-fa3390a5-7e01-4559-ae0c-eb15b38b22ca>" \
  --data "$(cat rabbitmq.conf | base64 --wrap=0)"
```

Perintah ini mengembalikan respon mirip dengan contoh berikut.

```
{
    "Arn": "arn:aws:mq:us-west-2:123456789012:configuration:c-b600ac8e-8183-4f74-
a713-983e59f30e3d",
    "Created": "2025-07-17T16:57:04.520931+00:00",
    "Id": "c-b600ac8e-8183-4f74-a713-983e59f30e3d",
    "LatestRevision": {
        "Created": "2025-07-17T16:57:39.172000+00:00",
        "Revision": 2
    },
    "Name": "rabbitmq-ldap-config",
    "Warnings": []
}
```

4. Buat broker dengan konfigurasi LDAP yang Anda buat di Langkah 2 prosedur ini. Untuk melakukan ini, gunakan perintah `create-broker` AWS CLI seperti yang ditunjukkan pada contoh berikut. Dalam perintah ini, berikan ID konfigurasi dan nomor revisi yang Anda peroleh dalam tanggapan Langkah 1 dan 2 masing-masing. Misalnya, `c-fa3390a5-7e01-4559-ae0c-eb15b38b22ca` dan 2.

```
aws mq create-broker \
 --broker-name "rabbitmq-ldap-test-1" \
 --engine-type "RABBITMQ" \
 --engine-version "3.13" \
 --host-instance-type "mq.m7g.large" \
 --deployment-mode "CLUSTER_MULTI_AZ" \
 --logs '{"General": true}' \
 --publicly-accessible \
```

```
 --configuration '{"Id": "<c-fa3390a5-7e01-4559-ae0c-eb15b38b22ca>","Revision":
 <2>}'
```

Perintah ini mengembalikan respon mirip dengan contoh berikut.

```
{
    "BrokerArn": "arn:aws:mq:us-west-2:123456789012:broker:rabbitmq-ldap-
broker:b-2a1b5133-a10c-49d2-879b-8c176c34cf73",
    "BrokerId": "b-2a1b5133-a10c-49d2-879b-8c176c34cf73"
}
```

⚠ Pembatasan penamaan broker

Peran IAM yang dibuat oleh tumpukan CDK prasyarat membatasi nama broker untuk memulai. `rabbitmq-ldap-test` Pastikan nama broker Anda mengikuti pola ini atau peran IAM tidak akan memiliki izin untuk mengambil peran untuk resolusi ARN.

5. Verifikasi bahwa status broker bertransisi dari `CREATION_IN_PROGRESS` ke`RUNNING`, menggunakan perintah `describe-broker` AWS CLI seperti yang ditunjukkan pada contoh berikut. Dalam perintah ini, berikan ID broker yang Anda peroleh dalam hasil langkah sebelumnya Misalnya,`b-2a1b5133-a10c-49d2-879b-8c176c34cf73`.

```
aws mq describe-broker \
 --broker-id "<b-2a1b5133-a10c-49d2-879b-8c176c34cf73>"
```

Perintah ini mengembalikan respon mirip dengan contoh berikut. Respons berikut adalah versi singkat dari output lengkap yang dikembalikan `describe-broker` perintah. Tanggapan ini menunjukkan status broker dan strategi otentikasi yang digunakan untuk mengamankan broker. Dalam hal ini, strategi `config_managed` otentikasi menunjukkan bahwa broker menggunakan metode otentikasi LDAP.

```
{
"AuthenticationStrategy": "config_managed",
```

```
    ...,
    "BrokerState": "RUNNING",
    ...
}
```

6. Validasi akses RabbitMQ menggunakan salah satu pengguna pengujian yang dibuat oleh tumpukan CDK prasyarat

```
# FIXME: Replace ${RabbitMqLdapTestStack.ConsoleUserPasswordArn} with the actual
 ARN from your deployed prerequisite CDK stack outputs
CONSOLE_PASSWORD=$(aws secretsmanager get-secret-value \
  --secret-id ${RabbitMqLdapTestStack.ConsoleUserPasswordArn} \
  --query 'SecretString' --output text)

# FIXME: Replace BrokerConsoleURL with the actual ConsoleURL retrieved by
# calling describe-broker for the broker created above
# Call management API /api/overview (should succeed)
curl -u RabbitMqConsoleUser:$CONSOLE_PASSWORD \
  https://${BrokerConsoleURL}/api/overview

# Try to create a user (should fail - console user only has monitoring permissions)
curl -u RabbitMqConsoleUser:$CONSOLE_PASSWORD \
  -X PUT https://${BrokerConsoleURL}/api/users/testuser \
  -H "Content-Type: application/json" \
  -d '{"password":"testpass","tags":"management"}'
```

Menggunakan otentikasi HTTP dan otorisasi untuk Amazon MQ untuk RabbitMQ

Tutorial ini menjelaskan cara mengkonfigurasi otentikasi HTTP dan otorisasi untuk Amazon MQ Anda untuk broker RabbitMQ menggunakan server HTTP eksternal.

> **Note**
>
> Plugin otentikasi HTTP hanya tersedia untuk Amazon MQ untuk RabbitMQ versi 4 ke atas.

Di halaman ini

- Prasyarat untuk mengonfigurasi otentikasi dan otorisasi HTTP
- Mengkonfigurasi otentikasi HTTP di RabbitMQ menggunakan CLI AWS

Prasyarat untuk mengonfigurasi otentikasi dan otorisasi HTTP

Anda dapat mengatur AWS sumber daya yang diperlukan dalam tutorial ini dengan menerapkan tumpukan AWS CDK untuk Amazon MQ untuk integrasi otentikasi HTTP RabbitMQ.

Tumpukan CDK ini secara otomatis membuat semua AWS sumber daya yang diperlukan termasuk server otentikasi HTTP, sertifikat, dan peran IAM. Lihat paket README untuk daftar lengkap sumber daya yang dibuat oleh tumpukan.

Jika Anda menyiapkan sumber daya secara manual alih-alih menggunakan tumpukan CDK, pastikan Anda memiliki infrastruktur yang setara sebelum mengonfigurasi otentikasi HTTP di Amazon MQ Anda untuk broker RabbitMQ.

Prasyarat untuk mengatur Amazon MQ

AWS Versi CLI >= 2.28.23 untuk membuat penambahan nama pengguna dan kata sandi opsional selama pembuatan broker.

Mengkonfigurasi otentikasi HTTP di RabbitMQ menggunakan CLI AWS

Prosedur ini menggunakan AWS CLI untuk membuat dan mengkonfigurasi sumber daya yang diperlukan. Dalam prosedur berikut, pastikan untuk mengganti nilai placeholder dengan nilai aktualnya.

1. Buat konfigurasi baru menggunakan perintah `create-configuration` AWS CLI seperti yang ditunjukkan pada contoh berikut.

```
aws mq create-configuration \
  --name "rabbitmq-http-config" \
  --engine-type "RABBITMQ" \
  --engine-version "4.2"
```

   Perintah ini mengembalikan respon mirip dengan contoh berikut.

```
{
    "Arn": "arn:aws:mq:us-west-2:123456789012:configuration:c-fa3390a5-7e01-4559-
ae0c-eb15b38b22ca",
    "AuthenticationStrategy": "simple",
    "Created": "2025-07-17T16:03:01.759943+00:00",
    "Id": "c-fa3390a5-7e01-4559-ae0c-eb15b38b22ca",
    "LatestRevision": {
        "Created": "2025-07-17T16:03:01.759000+00:00",
        "Description": "Auto-generated default for rabbitmq-http-config on RabbitMQ
 4.2",
        "Revision": 1
    },
    "Name": "rabbitmq-http-config"
}
```

2. Buat file konfigurasi yang dipanggil `rabbitmq.conf` untuk menggunakan HTTP sebagai metode otentikasi dan otorisasi, seperti yang ditunjukkan pada contoh berikut. Ganti semua nilai placeholder dalam template (ditandai dengan`${...}`) dengan nilai aktual dari output tumpukan AWS CDK prasyarat yang diterapkan atau infrastruktur yang setara.

```
auth_backends.1 = cache
auth_backends.2 = http
auth_cache.cached_backend = http

# HTTP authentication settings
# For more information, see https://github.com/rabbitmq/rabbitmq-auth-backend-http

# FIXME: Replace the ${...} placeholders with actual values
# from your deployed prerequisite CDK stack outputs.
auth_http.http_method = post
auth_http.user_path = ${HttpServerUserPath}
auth_http.vhost_path = ${HttpServerVhostPath}
auth_http.resource_path = ${HttpServerResourcePath}
auth_http.topic_path = ${HttpServerTopicPath}

# TLS/HTTPS configuration
auth_http.ssl_options.verify = verify_peer
auth_http.ssl_options.sni = test.amazonaws.com
```

```
# AWS integration for secure credential retrieval
# For more information, see https://github.com/amazon-mq/rabbitmq-aws

# Replace the ${...} placeholders with actual ARN values
# from your deployed prerequisite CDK stack outputs.
aws.arns.assume_role_arn = ${AmazonMqAssumeRoleArn}
aws.arns.auth_http.ssl_options.cacertfile = ${CaCertArn}
```

3. Perbarui konfigurasi menggunakan perintah `update-configuration` AWS CLI. Gunakan ID konfigurasi dari Langkah 3.

```
aws mq update-configuration \
  --configuration-id "<c-fa3390a5-7e01-4559-ae0c-eb15b38b22ca>" \
  --data "$(cat rabbitmq.conf | base64 --wrap=0)"
```

Perintah ini mengembalikan respon mirip dengan contoh berikut.

```
{
    "Arn": "arn:aws:mq:us-west-2:123456789012:configuration:c-fa3390a5-7e01-4559-
ae0c-eb15b38b22ca",
    "Created": "2025-07-17T16:57:04.520931+00:00",
    "Id": "c-fa3390a5-7e01-4559-ae0c-eb15b38b22ca",
    "LatestRevision": {
        "Created": "2025-07-17T16:57:39.172000+00:00",
        "Revision": 2
    },
    "Name": "rabbitmq-http-config",
    "Warnings": []
}
```

4. Buat broker dengan konfigurasi HTTP. Gunakan ID konfigurasi dan nomor revisi dari langkah sebelumnya.

```
aws mq create-broker \
  --broker-name "rabbitmq-http-test-1" \
  --engine-type "RABBITMQ" \
```

```
  --engine-version "4.2" \
  --host-instance-type "mq.m7g.large" \
  --deployment-mode "SINGLE_INSTANCE" \
  --logs '{"General": true}' \
  --publicly-accessible \
  --configuration '{"Id": "<c-fa3390a5-7e01-4559-ae0c-eb15b38b22ca>","Revision":
 <2>}'
```

Perintah ini mengembalikan respon mirip dengan contoh berikut.

```
{
    "BrokerArn": "arn:aws:mq:us-west-2:123456789012:broker:rabbitmq-http-
test-1:b-2a1b5133-a10c-49d2-879b-8c176c34cf73",
    "BrokerId": "b-2a1b5133-a10c-49d2-879b-8c176c34cf73"
}
```

5. Verifikasi bahwa status broker bertransisi dari `CREATION_IN_PROGRESS` ke`RUNNING`, menggunakan perintah `describe-broker` AWS CLI.

```
aws mq describe-broker \
  --broker-id "<b-2a1b5133-a10c-49d2-879b-8c176c34cf73>"
```

Perintah ini mengembalikan respon mirip dengan contoh berikut. Strategi `config_managed` otentikasi menunjukkan bahwa broker menggunakan metode otentikasi HTTP.

```
{
    "AuthenticationStrategy": "config_managed",
    ...,
    "BrokerState": "RUNNING",
    ...
}
```

6. Validasi akses RabbitMQ menggunakan salah satu pengguna pengujian yang dibuat oleh tumpukan CDK prasyarat

```
# FIXME: Replace ${RabbitMqHttpAuthElbStack.ConsoleUserPasswordArn} with the actual
 ARN from your deployed prerequisite CDK stack outputs
CONSOLE_PASSWORD=$(aws secretsmanager get-secret-value \
  --secret-id ${RabbitMqHttpAuthElbStack.ConsoleUserPasswordArn} \
  --query 'SecretString' --output text)

# FIXME: Replace BrokerConsoleURL with the actual ConsoleURL retrieved by
# calling describe-broker for the broker created above
# Call management API /api/overview (should succeed)
curl -u RabbitMqConsoleUser:$CONSOLE_PASSWORD \
  https://${BrokerConsoleURL}/api/overview

# Try to create a vhost (should fail - console user only has management
 permissions)
curl -u RabbitMqConsoleUser:$CONSOLE_PASSWORD \
  -X PUT https://${BrokerConsoleURL}/api/vhosts/test-vhost \
  -H "Content-Type: application/json" \
  -d '{}'
```

Menggunakan otentikasi sertifikat SSL untuk Amazon MQ untuk RabbitMQ

Tutorial ini menjelaskan cara mengonfigurasi otentikasi sertifikat SSL untuk Amazon MQ Anda untuk broker RabbitMQ menggunakan otoritas sertifikat pribadi.

> **Note**
>
> Plugin otentikasi sertifikat SSL hanya tersedia untuk Amazon MQ untuk RabbitMQ versi 4 ke atas.

Di halaman ini

- Prasyarat untuk mengonfigurasi otentikasi sertifikat SSL
- Mengkonfigurasi otentikasi sertifikat SSL di RabbitMQ menggunakan CLI AWS

Prasyarat untuk mengonfigurasi otentikasi sertifikat SSL

Otentikasi sertifikat SSL menggunakan TLS bersama (mTLS) untuk mengautentikasi klien menggunakan sertifikat X.509. Anda dapat mengatur AWS sumber daya yang diperlukan dalam tutorial ini dengan menerapkan tumpukan AWS CDK untuk Amazon MQ untuk integrasi MTLS RabbitMQ.

Tumpukan CDK ini secara otomatis membuat semua AWS sumber daya yang diperlukan termasuk otoritas sertifikat, sertifikat klien, dan peran IAM. Lihat paket README untuk daftar lengkap sumber daya yang dibuat oleh tumpukan.

> **Note**
>
> Sebelum menerapkan tumpukan CDK, atur variabel `RABBITMQ_TEST_USER_NAME` lingkungan. Nilai ini akan digunakan sebagai Nama Umum (CN) dalam sertifikat klien dan harus sesuai dengan nama pengguna yang Anda gunakan dalam langkah-langkah tutorial. Misalnya: `export RABBITMQ_TEST_USER_NAME="myuser"`

Jika Anda menyiapkan sumber daya secara manual alih-alih menggunakan tumpukan CDK, pastikan Anda memiliki infrastruktur yang setara sebelum mengonfigurasi otentikasi sertifikat SSL di Amazon MQ Anda untuk broker RabbitMQ.

Prasyarat untuk mengatur Amazon MQ

AWS Versi CLI >= 2.28.23 untuk membuat penambahan nama pengguna dan kata sandi opsional selama pembuatan broker.

Mengkonfigurasi otentikasi sertifikat SSL di RabbitMQ menggunakan CLI AWS

Prosedur ini menggunakan AWS CLI untuk membuat dan mengkonfigurasi sumber daya yang diperlukan. Dalam prosedur berikut, pastikan untuk mengganti nilai placeholder, seperti ConfigurationId dan Revision, `<c-fa3390a5-7e01-4559-ae0c-eb15b38b22ca>` dan`<2>`, dengan nilai sebenarnya.

1. Buat konfigurasi baru menggunakan perintah `create-configuration` AWS CLI seperti yang ditunjukkan pada contoh berikut.

```
aws mq create-configuration \
```

```
  --name "rabbitmq-ssl-config" \
  --engine-type "RABBITMQ" \
  --engine-version "4.2"
```

Perintah ini mengembalikan respon mirip dengan contoh berikut.

```
{
    "Arn": "arn:aws:mq:us-west-2:123456789012:configuration:c-fa3390a5-7e01-4559-
ae0c-eb15b38b22ca",
    "AuthenticationStrategy": "simple",
    "Created": "2025-07-17T16:03:01.759943+00:00",
    "Id": "c-fa3390a5-7e01-4559-ae0c-eb15b38b22ca",
    "LatestRevision": {
        "Created": "2025-07-17T16:03:01.759000+00:00",
        "Description": "Auto-generated default for rabbitmq-ssl-config on RabbitMQ
 4.2",
        "Revision": 1
    },
    "Name": "rabbitmq-ssl-config"
}
```

2. Buat file konfigurasi yang dipanggil `rabbitmq.conf` untuk menggunakan otentikasi sertifikat SSL, seperti yang ditunjukkan pada contoh berikut. Ganti semua nilai placeholder dalam template (ditandai dengan`${...}`) dengan nilai aktual dari output tumpukan AWS CDK prasyarat yang diterapkan atau infrastruktur yang setara.

```
auth_mechanisms.1 = EXTERNAL
ssl_cert_login_from = common_name

auth_backends.1 = internal

# Reject if no client cert
ssl_options.verify = verify_peer
ssl_options.fail_if_no_peer_cert = true

# AWS integration for secure credential retrieval
# For more information, see https://github.com/amazon-mq/rabbitmq-aws
```

```
# FIXME: Replace the ${...} placeholders with actual ARN values
# from your deployed prerequisite CDK stack outputs.
aws.arns.assume_role_arn = ${AmazonMqAssumeRoleArn}
aws.arns.ssl_options.cacertfile = ${CaCertArn}
```

3. Perbarui konfigurasi menggunakan perintah `update-configuration` AWS CLI seperti yang ditunjukkan pada contoh berikut. Dalam perintah ini, tambahkan ID konfigurasi yang Anda terima sebagai respons Langkah 1 dari prosedur ini. Misalnya, `c-fa3390a5-7e01-4559-ae0c-eb15b38b22ca`.

```
aws mq update-configuration \
  --configuration-id "<c-fa3390a5-7e01-4559-ae0c-eb15b38b22ca>" \
  --data "$(cat rabbitmq.conf | base64 --wrap=0)"
```

Perintah ini mengembalikan respon mirip dengan contoh berikut.

```
{
    "Arn": "arn:aws:mq:us-west-2:123456789012:configuration:c-fa3390a5-7e01-4559-
ae0c-eb15b38b22ca",
    "Created": "2025-07-17T16:57:04.520931+00:00",
    "Id": "c-fa3390a5-7e01-4559-ae0c-eb15b38b22ca",
    "LatestRevision": {
        "Created": "2025-07-17T16:57:39.172000+00:00",
        "Revision": 2
    },
    "Name": "rabbitmq-ssl-config",
    "Warnings": []
}
```

4. Buat broker dengan konfigurasi otentikasi sertifikat SSL yang Anda buat di Langkah 2 prosedur ini. Untuk melakukan ini, gunakan perintah `create-broker` AWS CLI seperti yang ditunjukkan pada contoh berikut. Dalam perintah ini, berikan ID konfigurasi dan nomor revisi yang Anda peroleh dalam tanggapan Langkah 1 dan 2 masing-masing. Misalnya, `c-fa3390a5-7e01-4559-ae0c-eb15b38b22ca` dan 2.

```
aws mq create-broker \
  --broker-name "rabbitmq-ssl-test-1" \
  --engine-type "RABBITMQ" \
  --engine-version "4.2" \
  --host-instance-type "mq.m7g.large" \
  --deployment-mode "SINGLE_INSTANCE" \
  --logs '{"General": true}' \
  --publicly-accessible \
  --configuration '{"Id": "<c-fa3390a5-7e01-4559-ae0c-eb15b38b22ca>","Revision":
 <2>}' \
  --users '[{"Username":"testuser","Password":"testpassword"}]'
```

Perintah ini mengembalikan respon mirip dengan contoh berikut.

```
{
    "BrokerArn": "arn:aws:mq:us-west-2:123456789012:broker:rabbitmq-ssl-
test-1:b-2a1b5133-a10c-49d2-879b-8c176c34cf73",
    "BrokerId": "b-2a1b5133-a10c-49d2-879b-8c176c34cf73"
}
```

5. Verifikasi bahwa status broker bertransisi dari `CREATION_IN_PROGRESS` ke`RUNNING`, menggunakan perintah `describe-broker` AWS CLI seperti yang ditunjukkan pada contoh berikut. Dalam perintah ini, berikan ID broker yang Anda peroleh pada hasil langkah sebelumnya. Misalnya, `b-2a1b5133-a10c-49d2-879b-8c176c34cf73`.

```
aws mq describe-broker \
  --broker-id "<b-2a1b5133-a10c-49d2-879b-8c176c34cf73>"
```

Perintah ini mengembalikan respon mirip dengan contoh berikut. Respons berikut adalah versi singkat dari output lengkap yang dikembalikan `describe-broker` perintah. Tanggapan ini menunjukkan status broker dan strategi otentikasi yang digunakan untuk mengamankan broker. Dalam hal ini, strategi `config_managed` otentikasi menunjukkan bahwa broker menggunakan metode otentikasi sertifikat SSL.

```
{
```

```
    "AuthenticationStrategy": "config_managed",
    ...,
    "BrokerState": "RUNNING",
    ...
}
```

6. Verifikasi otentikasi sertifikat SSL dengan skrip berikut`ssl.sh`.

   Gunakan skrip bash ini untuk menguji konektivitas ke Amazon MQ Anda untuk broker RabbitMQ. Skrip ini menggunakan sertifikat klien Anda untuk otentikasi dan memverifikasi apakah koneksi telah dikonfigurasi dengan benar. Jika berhasil dikonfigurasi, Anda akan melihat broker Anda mempublikasikan dan menggunakan pesan.

   Jika Anda menerima `ACCESS_REFUSED` kesalahan, Anda dapat memecahkan masalah pengaturan konfigurasi Anda dengan menggunakan CloudWatch log untuk broker Anda. Anda dapat menemukan tautan untuk grup CloudWatch log untuk broker Anda di konsol Amazon MQ.

   Dalam skrip ini, Anda harus memberikan nilai-nilai berikut:

   - `USERNAME`Nama umum (CN) dari sertifikat klien Anda.
   - `CLIENT_KEYSTORE`: Jalur ke file keystore klien Anda (PKCS12 format). Jika Anda menggunakan tumpukan CDK prasyarat, jalur defaultnya adalah. `$(pwd)/certs/client-keystore.p12`
   - `KEYSTORE_PASSWORD`: Kata sandi untuk keystore klien Anda. Jika Anda menggunakan tumpukan CDK prasyarat, kata sandi defaultnya adalah. `changeit`
   - `BROKER_DNS`: Anda dapat menemukan nilai ini di bawah Koneksi di halaman detail broker konsol Amazon MQ.

```
#! /bin/bash
set -e

# Client information
## FIXME: Update this value with the client ID and secret of your confidential
 application client
USERNAME=<client_cert_common_name>
CLIENT_KEYSTORE=$(pwd)/certs/client-keystore.p12
KEYSTORE_PASSWORD=changeit

BROKER_DNS=<broker_dns>
```

```
CONNECTION_STRING=amqps://${BROKER_DNS}:5671

# Produce/consume messages using the above connection string
QUEUES_COUNT=1
PRODUCERS_COUNT=1
CONSUMERS_COUNT=1
PRODUCER_RATE=1

finch run --rm --ulimit nofile=40960:40960 \
    -v ${CLIENT_KEYSTORE}:/certs/client-keystore.p12:ro \
    -e JAVA_TOOL_OPTIONS="-Djavax.net.ssl.keyStore=/certs/client-
keystore.p12 -Djavax.net.ssl.keyStorePassword=${KEYSTORE_PASSWORD} -
Djavax.net.ssl.keyStoreType=PKCS12" \
    pivotalrabbitmq/perf-test:latest \
    --queue-pattern 'test-queue-cert-%d' --queue-pattern-from 1 --queue-pattern-to
 $QUEUES_COUNT \
    --producers $PRODUCERS_COUNT --consumers $CONSUMERS_COUNT \
    --id "cert-test${QUEUES_COUNT}q${PRODUCERS_COUNT}p${CONSUMERS_COUNT}c
${PRODUCER_RATE}r" \
    --uri ${CONNECTION_STRING} \
    --sasl-external \
    --use-default-ssl-context \
    --flag persistent --rate $PRODUCER_RATE
```

Menggunakan mTL untuk AMQP dan endpoint manajemen

Tutorial ini menjelaskan cara mengkonfigurasi TLS bersama (mTLS) untuk koneksi klien AMQP dan antarmuka manajemen RabbitMQ menggunakan otoritas sertifikat pribadi.

> **Note**
>
> Penggunaan otoritas sertifikat swasta untuk mTLS hanya tersedia untuk Amazon MQ untuk RabbitMQ versi 4 ke atas.

Di halaman ini

- Prasyarat untuk mengkonfigurasi mTL
- Mengkonfigurasi MTL di RabbitMQ menggunakan CLI AWS

Prasyarat untuk mengkonfigurasi mTL

Anda dapat mengatur AWS sumber daya yang diperlukan dalam tutorial ini dengan menerapkan tumpukan AWS CDK untuk Amazon MQ untuk integrasi RabbitMQ MTLS dengan.

Tumpukan CDK ini secara otomatis membuat semua AWS sumber daya yang diperlukan termasuk otoritas sertifikat, sertifikat klien, dan peran IAM. Lihat paket README untuk daftar lengkap sumber daya yang dibuat oleh tumpukan.

Jika Anda menyiapkan sumber daya secara manual alih-alih menggunakan tumpukan CDK, pastikan Anda memiliki infrastruktur yang setara sebelum mengonfigurasi mTL di Amazon MQ Anda untuk broker RabbitMQ.

Prasyarat untuk mengatur Amazon MQ

AWS Versi CLI >= 2.28.23 untuk membuat penambahan nama pengguna dan kata sandi opsional selama pembuatan broker.

Mengkonfigurasi MTL di RabbitMQ menggunakan CLI AWS

Prosedur ini menggunakan AWS CLI untuk membuat dan mengkonfigurasi sumber daya yang diperlukan. Dalam prosedur berikut, pastikan untuk mengganti nilai placeholder, seperti ConfigurationId dan Revision, `<c-fa3390a5-7e01-4559-ae0c-eb15b38b22ca>` dan`<2>`, dengan nilai sebenarnya.

1. Buat konfigurasi baru menggunakan perintah `create-configuration` AWS CLI seperti yang ditunjukkan pada contoh berikut.

```
aws mq create-configuration \
  --name "rabbitmq-mtls-config" \
  --engine-type "RABBITMQ" \
  --engine-version "4.2"
```

   Perintah ini mengembalikan respon mirip dengan contoh berikut.

```
{
    "Arn": "arn:aws:mq:us-west-2:123456789012:configuration:c-fa3390a5-7e01-4559-
ae0c-eb15b38b22ca",
```

```
    "AuthenticationStrategy": "simple",
    "Created": "2025-07-17T16:03:01.759943+00:00",
    "Id": "c-fa3390a5-7e01-4559-ae0c-eb15b38b22ca",
    "LatestRevision": {
        "Created": "2025-07-17T16:03:01.759000+00:00",
        "Description": "Auto-generated default for rabbitmq-mtls-config on RabbitMQ
 4.2",
        "Revision": 1
    },
    "Name": "rabbitmq-mtls-config"
}
```

2. Buat file konfigurasi yang dipanggil `rabbitmq.conf` untuk mengkonfigurasi mTL untuk AMQP dan endpoint manajemen, seperti yang ditunjukkan pada contoh berikut. Ganti semua nilai placeholder dalam template (ditandai dengan`${...}`) dengan nilai aktual dari output tumpukan AWS CDK prasyarat yang diterapkan atau infrastruktur yang setara.

```
auth_backends.1 = internal

# TLS configuration
ssl_options.verify = verify_peer
ssl_options.fail_if_no_peer_cert = true
management.ssl.verify = verify_peer

# AWS integration for secure credential retrieval
# For more information, see https://github.com/amazon-mq/rabbitmq-aws

# FIXME: Replace the ${...} placeholders with actual ARN values
# from your deployed prerequisite CDK stack outputs.
aws.arns.assume_role_arn = ${AmazonMqAssumeRoleArn}
aws.arns.ssl_options.cacertfile = ${CaCertArn}
aws.arns.management.ssl.cacertfile = ${CaCertArn}
```

3. Perbarui konfigurasi menggunakan perintah `update-configuration` AWS CLI seperti yang ditunjukkan pada contoh berikut. Dalam perintah ini, tambahkan ID konfigurasi yang Anda terima sebagai respons Langkah 1 dari prosedur ini. Misalnya, `c-fa3390a5-7e01-4559-ae0c-eb15b38b22ca`.

```
aws mq update-configuration \
  --configuration-id "<c-fa3390a5-7e01-4559-ae0c-eb15b38b22ca>" \
  --data "$(cat rabbitmq.conf | base64 --wrap=0)"
```

Perintah ini mengembalikan respon mirip dengan contoh berikut.

```
{
    "Arn": "arn:aws:mq:us-west-2:123456789012:configuration:c-fa3390a5-7e01-4559-
ae0c-eb15b38b22ca",
    "Created": "2025-07-17T16:57:04.520931+00:00",
    "Id": "c-fa3390a5-7e01-4559-ae0c-eb15b38b22ca",
    "LatestRevision": {
        "Created": "2025-07-17T16:57:39.172000+00:00",
        "Revision": 2
    },
    "Name": "rabbitmq-mtls-config",
    "Warnings": []
}
```

4. Buat broker dengan konfigurasi mTLS yang Anda buat di Langkah 2 dari prosedur ini. Untuk melakukan ini, gunakan perintah `create-broker` AWS CLI seperti yang ditunjukkan pada contoh berikut. Dalam perintah ini, berikan ID konfigurasi dan nomor revisi yang Anda peroleh dalam tanggapan Langkah 1 dan 2 masing-masing. Misalnya, `c-fa3390a5-7e01-4559-ae0c-eb15b38b22ca` dan 2.

```
aws mq create-broker \
  --broker-name "rabbitmq-mtls-test-1" \
  --engine-type "RABBITMQ" \
  --engine-version "4.2" \
  --host-instance-type "mq.m7g.large" \
  --deployment-mode "SINGLE_INSTANCE" \
  --logs '{"General": true}' \
  --publicly-accessible \
  --configuration '{"Id": "<c-fa3390a5-7e01-4559-ae0c-eb15b38b22ca>","Revision":
 <2>}' \
  --users '[{"Username":"testuser","Password":"testpassword"}]'
```

Perintah ini mengembalikan respon mirip dengan contoh berikut.

```
{
    "BrokerArn": "arn:aws:mq:us-west-2:123456789012:broker:rabbitmq-mtls-
test-1:b-2a1b5133-a10c-49d2-879b-8c176c34cf73",
    "BrokerId": "b-2a1b5133-a10c-49d2-879b-8c176c34cf73"
}
```

5. Verifikasi bahwa status broker bertransisi dari `CREATION_IN_PROGRESS` ke`RUNNING`, menggunakan perintah `describe-broker` AWS CLI seperti yang ditunjukkan pada contoh berikut. Dalam perintah ini, berikan ID broker yang Anda peroleh pada hasil langkah sebelumnya. Misalnya, `b-2a1b5133-a10c-49d2-879b-8c176c34cf73`.

```
aws mq describe-broker \
  --broker-id "<b-2a1b5133-a10c-49d2-879b-8c176c34cf73>"
```

Perintah ini mengembalikan respon mirip dengan contoh berikut. Respons berikut adalah versi singkat dari output lengkap yang dikembalikan `describe-broker` perintah.

```
{
    "AuthenticationStrategy": "simple",
    ...,
    "BrokerState": "RUNNING",
    ...
}
```

6. Verifikasi otentikasi mTLS dengan skrip berikut`mtls.sh`.

Gunakan skrip bash ini untuk menguji konektivitas ke Amazon MQ Anda untuk broker RabbitMQ. Skrip ini menggunakan sertifikat klien Anda untuk mengautentikasi dan memverifikasi apakah koneksi telah dikonfigurasi dengan benar. Jika berhasil dikonfigurasi, Anda akan melihat broker Anda mempublikasikan dan menggunakan pesan.

Jika Anda menerima `ACCESS_REFUSED` kesalahan, Anda dapat memecahkan masalah pengaturan konfigurasi Anda dengan menggunakan CloudWatch log untuk broker Anda. Anda dapat menemukan tautan untuk grup CloudWatch log untuk broker Anda di konsol Amazon MQ.

Dalam skrip ini, Anda harus memberikan nilai-nilai berikut:

- `USERNAME`dan`PASSWORD`: Kredensyal pengguna RabbitMQ yang Anda buat dengan broker.
- `CLIENT_KEYSTORE`: Jalur ke file keystore klien Anda (PKCS12 format). Jika Anda menggunakan tumpukan CDK prasyarat, jalur defaultnya adalah. `$(pwd)/certs/client-keystore.p12`
- `KEYSTORE_PASSWORD`: Kata sandi untuk keystore klien Anda. Jika Anda menggunakan tumpukan CDK prasyarat, kata sandi defaultnya adalah. `changeit`
- `BROKER_DNS`: Anda dapat menemukan nilai ini di bawah Koneksi di halaman detail broker konsol Amazon MQ.

```
#! /bin/bash
set -e

# Client information
## FIXME: Update this value with the client ID and secret of your confidential
 application client
USERNAME=<testuser>
PASSWORD=<testpassword>
CLIENT_KEYSTORE=$(pwd)/certs/client-keystore.p12
KEYSTORE_PASSWORD=changeit

BROKER_DNS=<broker_dns>
CONNECTION_STRING=amqps://${USERNAME}:${PASSWORD}@${BROKER_DNS}:5671

# Produce/consume messages using the above connection string
QUEUES_COUNT=1
PRODUCERS_COUNT=1
CONSUMERS_COUNT=1
PRODUCER_RATE=1

finch run --rm --ulimit nofile=40960:40960 \
    -v ${CLIENT_KEYSTORE}:/certs/client-keystore.p12:ro \
```

```
    -e JAVA_TOOL_OPTIONS="-Djavax.net.ssl.keyStore=/certs/client-
keystore.p12 -Djavax.net.ssl.keyStorePassword=${KEYSTORE_PASSWORD} -
Djavax.net.ssl.keyStoreType=PKCS12" \
    pivotalrabbitmq/perf-test:latest \
    --queue-pattern 'test-queue-cert-%d' --queue-pattern-from 1 --queue-pattern-to
 $QUEUES_COUNT \
    --producers $PRODUCERS_COUNT --consumers $CONSUMERS_COUNT \
    --id "cert-test${QUEUES_COUNT}q${PRODUCERS_COUNT}p${CONSUMERS_COUNT}c
${PRODUCER_RATE}r" \
    --uri ${CONNECTION_STRING} \
    --use-default-ssl-context \
    --flag persistent --rate $PRODUCER_RATE
```

Menghubungkan aplikasi JMS Anda

Tutorial ini menunjukkan cara menghubungkan aplikasi JMS Anda ke Amazon MQ untuk broker RabbitMQ menggunakan klien RabbitMQ JMS. Anda akan belajar cara membuat produser untuk mengirim pesan dan konsumen untuk menerima pesan dari antrian RabbitMQ.

Sebelum Anda mulai, tambahkan dependensi RabbitMQ JMS yang sesuai ke proyek Maven Anda:

Untuk JMS 1.1 dan 2.0:

```
<dependencies>

  <dependency>
    <groupId>com.rabbitmq.jms</groupId>
    <artifactId>rabbitmq-jms</artifactId>
    <version>2.12.0</version>
  </dependency>

</dependencies>
```

Untuk JMS 3.1:

```
<dependencies>

  <dependency>
    <groupId>com.rabbitmq.jms</groupId>
    <artifactId>rabbitmq-jms</artifactId>
    <version>3.5.0</version>
  </dependency>
```

```
</dependencies>
```

Buat produser

Contoh kode berikut menunjukkan cara menulis ke antrian RabbitMQ menggunakan JMS:

```
import jakarta.jms.*;
import com.rabbitmq.jms.admin.*;

// Setting the connection factory
RMQConnectionFactory factory = new RMQConnectionFactory();
factory.setHost(envProps.getProperty("RABBITMQ_HOST", "localhost"));
factory.setPort(Integer.parseInt(envProps.getProperty("RABBITMQ_PORT", "5672")));
factory.setUsername(envProps.getProperty("RABBITMQ_USERNAME", "guest"));
factory.setPassword(envProps.getProperty("RABBITMQ_PASSWORD", "guest"));
factory.setVirtualHost(envProps.getProperty("RABBITMQ_VIRTUAL_HOST", "/"));
factory.useSslProtocol();

connection = factory.createConnection();
connection.start();

String queueName = "test-queue-jms";
Session session = connection.createSession(false, Session.AUTO_ACKNOWLEDGE);

RMQDestination destination = new RMQDestination(queueName, true, false);

// Send the message to the queue
MessageProducer producer = session.createProducer(destination);
producer.setDeliveryMode(DeliveryMode.PERSISTENT);

String msg_content = "Hello World!!";
TextMessage textMessage = session.createTextMessage(msg_content);
producer.send(textMessage);

System.out.printf("Published to AMQP queue '%s': %s", queueName, msg_content);
```

Ciptakan konsumen

Contoh kode berikut menunjukkan cara membaca dari antrian RabbitMQ menggunakan JMS:

```
import jakarta.jms.*;
import com.rabbitmq.jms.admin.*;
```

```
// Setting the connection factory
RMQConnectionFactory factory = new RMQConnectionFactory();
factory.setHost(envProps.getProperty("RABBITMQ_HOST", "localhost"));
factory.setPort(Integer.parseInt(envProps.getProperty("RABBITMQ_PORT", "5672")));
factory.setUsername(envProps.getProperty("RABBITMQ_USERNAME", "guest"));
factory.setPassword(envProps.getProperty("RABBITMQ_PASSWORD", "guest"));
factory.setVirtualHost(envProps.getProperty("RABBITMQ_VIRTUAL_HOST", "/"));
factory.useSslProtocol();

// Establish the connection and session
jakarta.jms.Connection connection = factory.createConnection();

String queueName = "test-queue-jms";
Session session = connection.createSession(false, Session.AUTO_ACKNOWLEDGE);

RMQDestination destination = new RMQDestination();
destination.setDestinationName(queueName);
destination.setAmqp(true);
destination.setAmqpQueueName(queueName);

// Initialize consumer
MessageConsumer consumer = session.createConsumer(destination);
consumer.setMessageListener(message -> {
    try {
        if (message instanceof TextMessage) {
            TextMessage textMessage = (TextMessage) message;
            System.out.printf("Message: %s%n", textMessage.getText());
        } else if (message instanceof BytesMessage) {
            BytesMessage bytesMessage = (BytesMessage) message;
            byte[] bytes = new byte[(int) bytesMessage.getBodyLength()];
            bytesMessage.readBytes(bytes);
            String content = new String(bytes);
            System.out.printf("Message: %s%n", content);
        } else {
            System.out.printf("Message: [%s]%n", message.getClass().getSimpleName());
        }
    } catch (JMSException e) {
        System.err.printf("Error processing message: %s%n", e.getMessage());
    }
});

connection.start();
```

Keamanan di Amazon MQ

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

Keamanan adalah tanggung jawab bersama antara Anda AWS dan Anda. Model tanggung jawab bersama menjelaskan hal ini sebagai keamanan dari cloud dan keamanan dalam cloud:

- Keamanan cloud — AWS bertanggung jawab untuk melindungi infrastruktur yang menjalankan AWS layanan di AWS Cloud. AWS juga memberi Anda layanan yang dapat Anda gunakan dengan aman. Auditor pihak ketiga secara teratur menguji dan memverifikasi efektivitas keamanan kami sebagai bagian dari Program AWS Kepatuhan Program AWS Kepatuhan . Untuk mempelajari tentang program kepatuhan yang berlaku untuk Amazon MQ, lihat AWS Layanan dalam Lingkup berdasarkan AWS Layanan Program Kepatuhan dalam Lingkup oleh Program .
- Keamanan di cloud — Tanggung jawab Anda ditentukan oleh AWS layanan yang Anda gunakan. Anda juga bertanggung jawab atas faktor lain, yang mencakup sensitivitas data Anda, persyaratan perusahaan Anda, serta undang-undang dan peraturan yang berlaku.

Dokumentasi ini membantu Anda memahami cara menerapkan model tanggung jawab bersama saat menggunakan Amazon MQ. Topik berikut menunjukkan kepada Anda cara mengonfigurasi Amazon MQ untuk memenuhi tujuan keamanan dan kepatuhan Anda. Anda juga mempelajari cara menggunakan AWS layanan lain yang membantu Anda memantau dan mengamankan sumber daya Amazon MQ Anda.

Topik

- Perlindungan data di Amazon MQ
- Identity and Access Management untuk Amazon MQ
- Validasi kepatuhan untuk Amazon MQ
- Ketahanan di Amazon MQ
- Keamanan infrastruktur di Amazon MQ
- Praktik terbaik keamanan untuk Amazon MQ

Perlindungan data di Amazon MQ

Model tanggung jawab AWS bersama model berlaku untuk perlindungan data di Amazon MQ. Seperti yang dijelaskan dalam model AWS ini, bertanggung jawab untuk melindungi infrastruktur global yang menjalankan semua AWS Cloud. Anda bertanggung jawab untuk mempertahankan kendali atas konten yang di-host pada infrastruktur ini. Anda juga bertanggung jawab atas tugas-tugas konfigurasi dan manajemen keamanan untuk Layanan AWS yang Anda gunakan. Lihat informasi yang lebih lengkap tentang privasi data dalam Pertanyaan Umum Privasi Data. Lihat informasi tentang perlindungan data di Eropa di pos blog Model Tanggung Jawab Bersama dan GDPR AWS di Blog Keamanan AWS .

Untuk tujuan perlindungan data, kami menyarankan Anda melindungi Akun AWS kredensyal dan mengatur pengguna individu dengan AWS IAM Identity Center atau AWS Identity and Access Management (IAM). Dengan cara itu, setiap pengguna hanya diberi izin yang diperlukan untuk memenuhi tanggung jawab tugasnya. Kami juga menyarankan supaya Anda mengamankan data dengan cara-cara berikut:

- Gunakan autentikasi multi-faktor (MFA) pada setiap akun.
- Gunakan SSL/TLS untuk berkomunikasi dengan AWS sumber daya. Kami mensyaratkan TLS 1.2 dan menganjurkan TLS 1.3.
- Siapkan API dan logging aktivitas pengguna dengan AWS CloudTrail. Untuk informasi tentang penggunaan CloudTrail jejak untuk menangkap AWS aktivitas, lihat Bekerja dengan CloudTrail jejak di AWS CloudTrail Panduan Pengguna.
- Gunakan solusi AWS enkripsi, bersama dengan semua kontrol keamanan default di dalamnya Layanan AWS.
- Gunakan layanan keamanan terkelola tingkat lanjut seperti Amazon Macie, yang membantu menemukan dan mengamankan data sensitif yang disimpan di Amazon S3.
- Jika Anda memerlukan modul kriptografi tervalidasi FIPS 140-3 saat mengakses AWS melalui antarmuka baris perintah atau API, gunakan titik akhir FIPS. Lihat informasi selengkapnya tentang titik akhir FIPS yang tersedia di Standar Pemrosesan Informasi Federal (FIPS) 140-3.

Kami sangat merekomendasikan agar Anda tidak pernah memasukkan informasi identifikasi yang sensitif, seperti nomor rekening pelanggan Anda, ke dalam tanda atau bidang isian bebas seperti bidang Nama. Ini termasuk saat Anda bekerja dengan Amazon MQ atau lainnya Layanan AWS menggunakan konsol, API AWS CLI, atau. AWS SDKs Data apa pun yang Anda masukkan ke dalam tanda atau bidang isian bebas yang digunakan untuk nama dapat digunakan untuk log penagihan

atau log diagnostik. Saat Anda memberikan URL ke server eksternal, kami sangat menganjurkan supaya Anda tidak menyertakan informasi kredensial di dalam URL untuk memvalidasi permintaan Anda ke server itu.

Untuk Amazon MQ untuk ActiveMQ dan Amazon MQ untuk broker RabbitMQ, jangan gunakan informasi identitas pribadi (PII) atau informasi rahasia atau sensitif lainnya untuk nama broker atau nama pengguna saat membuat sumber daya melalui konsol web broker, atau Amazon MQ API. Nama broker dan nama pengguna dapat diakses oleh AWS layanan lain, termasuk CloudWatch Log. Nama pengguna broker tidak dimaksudkan untuk digunakan untuk data pribadi atau sensitif.

> **Important**
>
> TLS 1.3 tidak tersedia untuk broker RabbitMQ.

Enkripsi

Data pengguna yang disimpan di Amazon MQ dienkripsi saat istirahat. Enkripsi Amazon MQ saat istirahat memberikan keamanan yang ditingkatkan dengan mengenkripsi data Anda menggunakan kunci enkripsi yang disimpan di AWS Key Management Service (KMS). Layanan ini membantu mengurangi beban operasional dan kompleksitas yang terlibat dalam melindungi data sensitif. Dengan enkripsi saat istirahat, Anda dapat membangun aplikasi yang sensitif terhadap keamanan yang memenuhi persyaratan kepatuhan enkripsi dan peraturan.

Semua koneksi antara broker Amazon MQ menggunakan Keamanan Lapisan Pengangkutan (TLS) untuk memberikan enkripsi dalam transit.

Amazon MQ mengenkripsi pesan saat istirahat dan dalam transit menggunakan kunci enkripsi yang dikelola dan disimpan dengan aman. Untuk informasi selengkapnya, lihat Panduan Developer AWS Encryption SDK.

Enkripsi saat diam

Amazon MQ terintegrasi dengan AWS Key Management Service (KMS) untuk menawarkan enkripsi sisi server yang transparan. Amazon MQ selalu mengenkripsi data at rest.

Saat Anda membuat Amazon MQ untuk broker ActiveMQ atau Amazon MQ untuk broker RabbitMQ, Anda dapat menentukan yang Anda AWS KMS key ingin Amazon MQ gunakan untuk mengenkripsi

data Anda saat istirahat. Jika Anda tidak menentukan kunci KMS, Amazon MQ membuat kunci KMS AWS yang dimiliki untuk Anda dan menggunakannya atas nama Anda. Amazon MQ saat ini mendukung kunci KMS simetris. Untuk informasi selengkapnya tentang kunci KMS, lihat AWS KMS keys.

Saat membuat broker, Anda dapat mengonfigurasi kunci yang digunakan Amazon MQ untuk kunci enkripsi Anda dengan memilih salah satu kunci berikut.

- Kunci KMS milik Amazon MQ (default) - Kunci dimiliki dan dikelola oleh Amazon MQ dan tidak ada di akun Anda.
- AWS kunci KMS AWS terkelola - Kunci KMS terkelola (`aws/mq`) adalah kunci KMS di akun Anda yang dibuat, dikelola, dan digunakan atas nama Anda oleh Amazon MQ.
- Pilih kunci KMS yang ada dan dikelola pelanggan — Kunci KMS yang dikelola pelanggan dibuat dan dikelola oleh Anda di AWS Key Management Service (KMS).

⚠ Important

- Mencabut hibah tidak dapat dibatalkan. Hapus broker untuk mencabut hak akses.
- Untuk Amazon MQ untuk broker ActiveMQ yang menggunakan Amazon Elastic File System (EFS) untuk menyimpan data pesan, mungkin diperlukan beberapa jam agar izin menggunakan kunci KMS di akun Anda dicabut setelah mengambil tindakan yang diperlukan.
- Untuk Amazon MQ untuk RabbitMQ dan Amazon MQ untuk broker ActiveMQ yang menggunakan EBS untuk menyimpan data pesan, jika Anda menonaktifkan, menjadwalkan penghapusan, atau mencabut hibah yang memberikan izin Amazon EBS untuk menggunakan kunci KMS di akun Anda, Amazon MQ tidak dapat mempertahankan broker Anda, dan dapat berubah menjadi status terdegradasi.
- Jika Anda telah menonaktifkan kunci atau menjadwalkan kunci yang akan dihapus, Anda dapat mengaktifkan kembali kunci atau membatalkan penghapusan kunci dan menjaga agar broker Anda tetap terjaga.
- Mungkin diperlukan jam server untuk menonaktifkan kunci atau mencabut hibah setelah mengambil tindakan yang diperlukan.
- Untuk mengenkripsi atau mendekripsi CloudWatch log, Anda tidak dapat mengonfigurasi apa yang digunakan Amazon MQ untuk kunci enkripsi Anda. CloudWatch log melindungi data saat istirahat menggunakan enkripsi, dan grup log dienkripsi. Layanan CloudWatch

log mengelola enkripsi sisi server secara default. Untuk informasi selengkapnya tentang cara grup log dienkripsi, lihat Panduan Pengguna CloudWatch Log Amazon.

Saat membuat broker instans tunggal dengan kunci KMS untuk RabbitMQ, Anda akan melihat dua `CreateGrant` peristiwa masuk. AWS CloudTrail Acara pertama adalah Amazon MQ membuat hibah untuk kunci KMS. Acara kedua adalah EBS membuat hibah untuk EBS untuk digunakan.

CreateGrant AWS CloudTrail entri log: broker contoh tunggal

mq_grant


```
{
    "eventVersion": "1.08",
    "userIdentity": {
        "type": "AssumedRole",
        "principalId": "AKIAIOSFODNN7EXAMPLE",
        "arn": "arn:aws:iam::111122223333:user/AmazonMqConsole",
        "accountId": "111122223333",
        "accessKeyId": "AKIAI44QH8DHBEXAMPLE",
        "sessionContext": {
            "sessionIssuer": {
                "type": "Role",
                "principalId": "AKIAIOSFODNN7EXAMPLE",
                "arn": "arn:aws:iam::111122223333:user/AmazonMqConsole",
                "accountId": "111122223333",
                "userName": "AmazonMqConsole"
            },
            "webIdFederationData": {},
            "attributes": {
                "creationDate": "2023-02-23T18:59:10Z",
                "mfaAuthenticated": "false"
            }
        },
        "invokedBy": "mq.amazonaws.com"
    },
    "eventTime": "2018-06-28T22:23:46Z",
    "eventSource": "amazonmq.amazonaws.com",
    "eventName": "CreateGrant",
    "awsRegion": "us-west-2",
    "sourceIPAddress": "203.0.113.0",
    "userAgent": "PostmanRuntime/7.1.5",
```

```
    "requestParameters": {
        "granteePrincipal": "mq.amazonaws.com",
        "keyId": "arn:aws:kms:us-east-1:316438333700:key/bdbe42ae-f825-4e78-
a8a1-828d411c4be2",
        "retiringPrincipal": "mq.amazonaws.com",
        "operations": [
            "CreateGrant",
            "Decrypt",
            "GenerateDataKeyWithoutPlaintext",
            "ReEncryptFrom",
            "ReEncryptTo",
            "DescribeKey"
        ]
    },
    "responseElements": {
        "grantId":
 "0ab0ac0d0b000f00ea00cc0a0e00fc00bce000c000f0000000c0bc0a0000aaafSAMPLE",
        "keyId": "arn:aws:kms:us-
west-2:111122223333:key/1234abcd-12ab-34cd-56ef-123456SAMPLE",

    "requestID": "ff000af-00eb-00ce-0e00-ea000fb0fba0SAMPLE",
    "eventID": "ff000af-00eb-00ce-0e00-ea000fb0fba0SAMPLE",
    "readOnly": false,
    "resources": [
        {
           "accountId": "111122223333",
            "type": "AWS::KMS::Key",
            "ARN": "arn:aws:kms:us-
west-2:111122223333:key/1234abcd-12ab-34cd-56ef-123456SAMPLE"
        }
    ],
    "eventType": "AwsApiCall",
    "managementEvent": true,
    "recipientAccountId": "111122223333",
    "eventCategory": "Management",
    "sessionCredentialFromConsole": "true"
}
```

EBS grant creation

Anda akan melihat satu acara untuk pembuatan hibah EBS.

```
                                    {
    "eventVersion": "1.08",
    "userIdentity": {
        "type": "AWSService",
        "invokedBy": "mq.amazonaws.com"
    },
    "eventTime": "2023-02-23T19:09:40Z",
    "eventSource": "kms.amazonaws.com",
    "eventName": "CreateGrant",
    "awsRegion": "us-east-1",
    "sourceIPAddress": "mq.amazonaws.com",
    "userAgent": "ExampleDesktop/1.0 (V1; OS)",
    "requestParameters": {
        "granteePrincipal": "mq.amazonaws.com",
        "keyId": "arn:aws:kms:us-
west-2:111122223333:key/1234abcd-12ab-34cd-56ef-123456SAMPLE",
        "constraints": {
            "encryptionContextSubset": {
                "aws:ebs:id": "vol-0b670f00f7d5417c0"
            }
        },
        "operations": [
            "Decrypt"
        ],
        "retiringPrincipal": "ec2.us-east-1.amazonaws.com"
    },
    "responseElements": {
        "grantId":
 "0ab0ac0d0b000f00ea00cc0a0e00fc00bce000c000f0000000c0bc0a0000aaafSAMPLE",
        "keyId": "arn:aws:kms:us-
west-2:111122223333:key/1234abcd-12ab-34cd-56ef-123456SAMPLE",
    },
    "requestID": "ff000af-00eb-00ce-0e00-ea000fb0fba0SAMPLE",
    "eventID": "ff000af-00eb-00ce-0e00-ea000fb0fba0SAMPLE",
    "readOnly": false,
    "resources": [
        {
            "accountId": "111122223333",
            "type": "AWS::KMS::Key",
            "ARN": "arn:aws:kms:us-
west-2:111122223333:key/1234abcd-12ab-34cd-56ef-123456SAMPLE"
        }
    ],
    "eventType": "AwsApiCall",
```

```
    "managementEvent": true,
    "recipientAccountId": "111122223333",
    "sharedEventID": "ff000af-00eb-00ce-0e00-ea000fb0fba0SAMPLE",
    "eventCategory": "Management"
}
```

Saat membuat penerapan cluster dengan kunci KMS untuk RabbitMQ, Anda akan melihat lima peristiwa masuk. `CreateGrant` AWS CloudTrail Dua acara pertama adalah kreasi hibah untuk Amazon MQ. Tiga acara berikutnya adalah hibah yang dibuat oleh EBS untuk digunakan EBS.

CreateGrant AWS CloudTrail entri log: penyebaran cluster

mq_grant

```
{
    "eventVersion": "1.08",
    "userIdentity": {
        "type": "AssumedRole",
        "principalId": "AKIAIOSFODNN7EXAMPLE",
        "arn": "arn:aws:iam::111122223333:user/AmazonMqConsole",
        "accountId": "111122223333",
        "accessKeyId": "AKIAI44QH8DHBEXAMPLE",
        "sessionContext": {
            "sessionIssuer": {
                "type": "Role",
                "principalId": "AKIAIOSFODNN7EXAMPLE",
                "arn": "arn:aws:iam::111122223333:user/AmazonMqConsole",
                "accountId": "111122223333",
                "userName": "AmazonMqConsole"
            },
            "webIdFederationData": {},
            "attributes": {
                "creationDate": "2023-02-23T18:59:10Z",
                "mfaAuthenticated": "false"
            }
        },
        "invokedBy": "mq.amazonaws.com"
    },
    "eventTime": "2018-06-28T22:23:46Z",
    "eventSource": "amazonmq.amazonaws.com",
```

```
    "eventName": "CreateGrant",
    "awsRegion": "us-west-2",
    "sourceIPAddress": "203.0.113.0",
    "userAgent": "PostmanRuntime/7.1.5",
    "requestParameters": {
        "granteePrincipal": "mq.amazonaws.com",
        "keyId": "arn:aws:kms:us-east-1:316438333700:key/bdbe42ae-f825-4e78-
a8a1-828d411c4be2",
        "retiringPrincipal": "mq.amazonaws.com",
        "operations": [
            "CreateGrant",
            "Encrypt",
            "Decrypt",
            "ReEncryptFrom",
            "ReEncryptTo",
            "GenerateDataKey",
            "GenerateDataKeyWithoutPlaintext",
            "DescribeKey"
        ]
    },
    "responseElements": {
        "grantId":
 "0ab0ac0d0b000f00ea00cc0a0e00fc00bce000c000f0000000c0bc0a0000aaafSAMPLE",
        "keyId": "arn:aws:kms:us-
west-2:111122223333:key/1234abcd-12ab-34cd-56ef-123456SAMPLE",

    "requestID": "ff000af-00eb-00ce-0e00-ea000fb0fba0SAMPLE",
    "eventID": "ff000af-00eb-00ce-0e00-ea000fb0fba0SAMPLE",
    "readOnly": false,
    "resources": [
        {
           "accountId": "111122223333",
            "type": "AWS::KMS::Key",
            "ARN": "arn:aws:kms:us-
west-2:111122223333:key/1234abcd-12ab-34cd-56ef-123456SAMPLE"
        }
    ],
    "eventType": "AwsApiCall",
    "managementEvent": true,
    "recipientAccountId": "111122223333",
    "eventCategory": "Management",
    "sessionCredentialFromConsole": "true"
}
```

mq_rabbit_grant


```
{
    "eventVersion": "1.08",
    "userIdentity": {
        "type": "AssumedRole",
        "principalId": "AKIAIOSFODNN7EXAMPLE",
        "arn": "arn:aws:iam::111122223333:user/AmazonMqConsole",
        "accountId": "111122223333",
        "accessKeyId": "AKIAI44QH8DHBEXAMPLE",
        "sessionContext": {
            "sessionIssuer": {
                "type": "Role",
                "principalId": "AKIAIOSFODNN7EXAMPLE",
                "arn": "arn:aws:iam::111122223333:user/AmazonMqConsole",
                "accountId": "111122223333",
                "userName": "AmazonMqConsole"
            },
            "webIdFederationData": {},
            "attributes": {
                "creationDate": "2023-02-23T18:59:10Z",
                "mfaAuthenticated": "false"
            }
        },
        "invokedBy": "mq.amazonaws.com"
    },
    "eventTime": "2018-06-28T22:23:46Z",
    "eventSource": "amazonmq.amazonaws.com",
    "eventName": "CreateGrant",
    "awsRegion": "us-west-2",
    "sourceIPAddress": "203.0.113.0",
    "userAgent": "PostmanRuntime/7.1.5",
    "requestParameters": {
        "granteePrincipal": "mq.amazonaws.com",
        "retiringPrincipal": "mq.amazonaws.com",
        "operations": [
            "DescribeKey"
        ],
        "keyId": "arn:aws:kms:us-
west-2:111122223333:key/1234abcd-12ab-34cd-56ef-123456SAMPLE",
```

```
    },
    "responseElements": {
        "grantId":
 "0ab0ac0d0b000f00ea00cc0a0e00fc00bce000c000f0000000c0bc0a0000aaafSAMPLE",
        "keyId": "arn:aws:kms:us-
west-2:111122223333:key/1234abcd-12ab-34cd-56ef-123456SAMPLE",

    "requestID": "ff000af-00eb-00ce-0e00-ea000fb0fba0SAMPLE",
    "eventID": "ff000af-00eb-00ce-0e00-ea000fb0fba0SAMPLE",
    "readOnly": false,
    "resources": [
        {
           "accountId": "111122223333",
            "type": "AWS::KMS::Key",
            "ARN": "arn:aws:kms:us-
west-2:111122223333:key/1234abcd-12ab-34cd-56ef-123456SAMPLE"
        }
    ],
    "eventType": "AwsApiCall",
    "managementEvent": true,
    "recipientAccountId": "111122223333",
    "eventCategory": "Management",
    "sessionCredentialFromConsole": "true"
}
```

EBS grant creation

Anda akan melihat tiga acara untuk pembuatan hibah EBS.

```
                                    {
    "eventVersion": "1.08",
    "userIdentity": {
        "type": "AWSService",
        "invokedBy": "mq.amazonaws.com"
    },
    "eventTime": "2023-02-23T19:09:40Z",
    "eventSource": "kms.amazonaws.com",
    "eventName": "CreateGrant",
    "awsRegion": "us-east-1",
    "sourceIPAddress": "mq.amazonaws.com",
    "userAgent": "ExampleDesktop/1.0 (V1; OS)",
    "requestParameters": {
```

```
        "granteePrincipal": "mq.amazonaws.com",
        "keyId": "arn:aws:kms:us-
west-2:111122223333:key/1234abcd-12ab-34cd-56ef-123456SAMPLE",
        "constraints": {
            "encryptionContextSubset": {
                "aws:ebs:id": "vol-0b670f00f7d5417c0"
            }
        },
        "operations": [
            "Decrypt"
        ],
        "retiringPrincipal": "ec2.us-east-1.amazonaws.com"
    },
    "responseElements": {
        "grantId":
 "0ab0ac0d0b000f00ea00cc0a0e00fc00bce000c000f0000000c0bc0a0000aaafSAMPLE",
        "keyId": "arn:aws:kms:us-
west-2:111122223333:key/1234abcd-12ab-34cd-56ef-123456SAMPLE",
    },
    "requestID": "ff000af-00eb-00ce-0e00-ea000fb0fba0SAMPLE",
    "eventID": "ff000af-00eb-00ce-0e00-ea000fb0fba0SAMPLE",
    "readOnly": false,
    "resources": [
        {
            "accountId": "111122223333",
            "type": "AWS::KMS::Key",
            "ARN": "arn:aws:kms:us-
west-2:111122223333:key/1234abcd-12ab-34cd-56ef-123456SAMPLE"
        }
    ],
    "eventType": "AwsApiCall",
    "managementEvent": true,
    "recipientAccountId": "111122223333",
    "sharedEventID": "ff000af-00eb-00ce-0e00-ea000fb0fba0SAMPLE",
    "eventCategory": "Management"
}
```


Untuk informasi selengkapnya tentang kunci KMS, lihat AWS KMS keys dalam Panduan Developer AWS Key Management Service .

Enkripsi saat bergerak

Amazon MQ untuk ActiveMQ: Amazon MQ untuk ActiveMQ memerlukan Transport Layer Security (TLS) yang kuat dan mengenkripsi data dalam perjalanan antara broker penyebaran Amazon MQ Anda. Semua data yang lewat antara broker Amazon MQ dienkripsi menggunakan Transport Layer Security (TLS) yang kuat. Ini berlaku untuk semua protokol yang tersedia.

Amazon MQ untuk RabbitMQ: Amazon MQ untuk RabbitMQ memerlukan enkripsi Transport Layer Security (TLS) yang kuat untuk semua koneksi klien. Lalu lintas replikasi cluster RabbitMQ hanya transit VPC broker Anda dan semua lalu lintas jaringan antara pusat AWS data dienkripsi secara transparan pada lapisan fisik. Amazon MQ untuk broker berkerumun RabbitMQ saat ini tidak mendukung enkripsi antar-node untuk replikasi cluster. Untuk mempelajari selengkapnya data-in-transit, lihat Mengenkripsi Data-at-Rest dan -Dalam Transit.

Protokol Amazon MQ for ActiveMQ

Anda dapat mengakses broker ActiveMQ menggunakan protokol berikut dengan TLS yang diaktifkan:

- AMQP
- MQTT
- MQTT lebih WebSocket
- OpenWire
- MENGINJAK
- STOMP berakhir WebSocket

Cipher Suite TLS yang didukung untuk ActiveMQ

ActiveMQ di Amazon MQ mendukung cipher suite berikut:

- TLS_ECDHE_RSA_WITH_AES_256_GCM_ SHA384
- TLS_ECDHE_RSA_WITH_AES_256_CBC_ SHA384
- TLS_ECDHE_RSA_WITH_AES_256_CBC_SHA
- TLS_DHE_RSA_WITH_AES_256_GCM_ SHA384
- TLS_DHE_RSA_DENGAN_AES_256_CBC_ SHA256
- TLS_DHE_RSA_WITH_AES_256_CBC_SHA

- TLS_RSA_WITH_AES_256_GCM_ SHA384
- TLS_RSA_WITH_AES_256_CBC_ SHA256
- TLS_RSA_WITH_AES_256_CBC_SHA
- TLS_ECDHE_RSA_WITH_AES_128_GCM_ SHA256
- TLS_ECDHE_RSA_WITH_AES_128_CBC_ SHA256
- TLS_ECDHE_RSA_WITH_AES_128_CBC_SHA
- TLS_DHE_RSA_WITH_AES_128_GCM_ SHA256
- TLS_DHE_RSA_DENGAN_AES_128_CBC_ SHA256
- TLS_DHE_RSA_WITH_AES_128_CBC_SHA
- TLS_RSA_WITH_AES_128_GCM_ SHA256
- TLS_RSA_DENGAN_AES_128_CBC_ SHA256
- TLS_RSA_WITH_AES_128_CBC_SHA

Protokol Amazon MQ for RabbitMQ

Anda dapat mengakses broker RabbitMQ menggunakan protokol berikut dengan TLS yang diaktifkan:

- AMQP (0-9-1)

Cipher Suite TLS yang didukung untuk RabbitMQ

RabbitMQ di Amazon MQ mendukung cipher suite berikut:

- TLS_ECDHE_RSA_WITH_AES_256_GCM_ SHA384
- TLS_ECDHE_RSA_WITH_AES_128_GCM_ SHA256

Identity and Access Management untuk Amazon MQ

AWS Identity and Access Management (IAM) adalah Layanan AWS yang membantu administrator mengontrol akses ke AWS sumber daya dengan aman. Administrator IAM mengontrol siapa yang dapat diautentikasi (masuk) dan diotorisasi (memiliki izin) untuk menggunakan sumber daya Amazon MQ. IAM adalah Layanan AWS yang dapat Anda gunakan tanpa biaya tambahan.

Topik

- Audiens
- Mengautentikasi dengan identitas
- Mengelola akses menggunakan kebijakan
- Cara kerja Amazon MQ dengan IAM
- Contoh kebijakan berbasis Identitas Amazon MQ
- Autentikasi dan otorisasi API untuk Amazon MQ
- Otentikasi dan otorisasi broker
- AWS kebijakan terkelola untuk Amazon MQ
- Menggunakan peran tertaut layanan untuk Amazon MQ
- Pemecahan masalah identitas dan akses Amazon MQ

Audiens

Cara Anda menggunakan AWS Identity and Access Management (IAM) berbeda berdasarkan peran Anda:

- Pengguna layanan - minta izin dari administrator Anda jika Anda tidak dapat mengakses fitur (lihat Pemecahan masalah identitas dan akses Amazon MQ)
- Administrator layanan - tentukan akses pengguna dan mengirimkan permintaan izin (lihat Cara kerja Amazon MQ dengan IAM)
- Administrator IAM - tulis kebijakan untuk mengelola akses (lihat Contoh kebijakan berbasis Identitas Amazon MQ)

Mengautentikasi dengan identitas

Otentikasi adalah cara Anda masuk AWS menggunakan kredensyal identitas Anda. Anda harus diautentikasi sebagai Pengguna root akun AWS, pengguna IAM, atau dengan mengasumsikan peran IAM.

Anda dapat masuk sebagai identitas federasi menggunakan kredensil dari sumber identitas seperti AWS IAM Identity Center (Pusat Identitas IAM), autentikasi masuk tunggal, atau kredensil. Google/ Facebook Untuk informasi selengkapnya tentang cara masuk, lihat Cara masuk ke Akun AWS Anda dalam Panduan Pengguna AWS Sign-In .

Untuk akses terprogram, AWS sediakan SDK dan CLI untuk menandatangani permintaan secara kriptografis. Untuk informasi selengkapnya, lihat AWS Signature Version 4 untuk permintaan API dalam Panduan Pengguna IAM.

Akun AWS pengguna root

Saat Anda membuat Akun AWS, Anda mulai dengan satu identitas masuk yang disebut pengguna Akun AWS root yang memiliki akses lengkap ke semua Layanan AWS dan sumber daya. Kami sangat menyarankan agar Anda tidak menggunakan pengguna root untuk tugas sehari-hari. Untuk tugas yang memerlukan kredensial pengguna root, lihat Tugas yang memerlukan kredensial pengguna root dalam Panduan Pengguna IAM.

Pengguna dan grup

Pengguna IAM adalah identitas dengan izin khusus untuk satu orang atau aplikasi. Sebaiknya gunakan kredensial sementara alih-alih pengguna IAM dengan kredensial jangka panjang. Untuk informasi selengkapnya, lihat Mewajibkan pengguna manusia untuk menggunakan federasi dengan penyedia identitas untuk mengakses AWS menggunakan kredensi sementara di Panduan Pengguna IAM.

Grup IAM menentukan kumpulan pengguna IAM dan mempermudah pengelolaan izin untuk pengguna dalam jumlah besar. Untuk mempelajari selengkapnya, lihat Kasus penggunaan untuk pengguna IAM dalam Panduan Pengguna IAM.

Peran IAM

Peran IAM adalah identitas dengan izin khusus yang menyediakan kredensial sementara. Anda dapat mengambil peran dengan beralih dari pengguna ke peran IAM (konsol) atau dengan memanggil operasi AWS CLI atau AWS API. Untuk informasi selengkapnya, lihat Metode untuk mengambil peran dalam Panduan Pengguna IAM.

Peran IAM berguna untuk akses pengguna terfederasi, izin pengguna IAM sementara, akses lintas akun, akses lintas layanan, dan aplikasi yang berjalan di Amazon EC2. Untuk informasi selengkapnya, lihat Akses sumber daya lintas akun di IAM dalam Panduan Pengguna IAM.

Mengelola akses menggunakan kebijakan

Anda mengontrol akses AWS dengan membuat kebijakan dan melampirkannya ke AWS identitas atau sumber daya. Kebijakan menentukan izin saat dikaitkan dengan identitas atau sumber daya. AWS mengevaluasi kebijakan ini ketika kepala sekolah membuat permintaan. Sebagian besar

kebijakan disimpan AWS sebagai dokumen JSON. Untuk informasi selengkapnya tentang dokumen kebijakan JSON, lihat Gambaran umum kebijakan JSON dalam Panduan Pengguna IAM.

Menggunakan kebijakan, administrator menentukan siapa yang memiliki akses ke apa dengan mendefinisikan principal mana yang dapat melakukan tindakan pada sumber daya apa, dan dalam kondisi apa.

Secara default, pengguna dan peran tidak memiliki izin. Administrator IAM membuat kebijakan IAM dan menambahkannya ke peran, yang kemudian dapat diambil oleh pengguna. Kebijakan IAM mendefinisikan izin terlepas dari metode yang Anda gunakan untuk melakukan operasinya.

Kebijakan berbasis identitas

Kebijakan berbasis identitas adalah dokumen kebijakan izin JSON yang Anda lampirkan ke identitas (pengguna, grup, atau peran). Kebijakan ini mengontrol tindakan apa yang bisa dilakukan oleh identitas tersebut, terhadap sumber daya yang mana, dan dalam kondisi apa. Untuk mempelajari cara membuat kebijakan berbasis identitas, lihat Tentukan izin IAM kustom dengan kebijakan yang dikelola pelanggan dalam Panduan Pengguna IAM.

Kebijakan berbasis identitas dapat berupa kebijakan inline (disematkan langsung ke dalam satu identitas) atau kebijakan terkelola (kebijakan mandiri yang dilampirkan pada banyak identitas). Untuk mempelajari cara memilih antara kebijakan terkelola dan kebijakan inline, lihat Pilih antara kebijakan terkelola dan kebijakan inline dalam Panduan Pengguna IAM.

Kebijakan berbasis sumber daya

Kebijakan berbasis sumber daya adalah dokumen kebijakan JSON yang Anda lampirkan ke sumber daya. Contohnya termasuk kebijakan kepercayaan 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. Anda harus menentukan principal dalam kebijakan berbasis sumber daya.

Kebijakan berbasis sumber daya merupakan kebijakan inline yang terletak di layanan tersebut. Anda tidak dapat menggunakan kebijakan AWS terkelola dari IAM dalam kebijakan berbasis sumber daya.

Daftar Kontrol Akses (ACLs)

Access control lists (ACLs) mengontrol prinsipal mana (anggota akun, pengguna, atau peran) yang memiliki izin untuk mengakses sumber daya. ACLs mirip dengan kebijakan berbasis sumber daya, meskipun mereka tidak menggunakan format dokumen kebijakan JSON.

Amazon S3, AWS WAF, dan Amazon VPC adalah contoh layanan yang mendukung. ACLs Untuk mempelajari selengkapnya ACLs, lihat Ringkasan daftar kontrol akses (ACL) di Panduan Pengembang Layanan Penyimpanan Sederhana Amazon.

Jenis-jenis kebijakan lain

AWS mendukung jenis kebijakan tambahan yang dapat menetapkan izin maksimum yang diberikan oleh jenis kebijakan yang lebih umum:

- Batasan izin – Menetapkan izin maksimum yang dapat diberikan oleh kebijakan berbasis identitas kepada entitas IAM. Untuk informasi selengkapnya, lihat Batasan izin untuk entitas IAM dalam Panduan Pengguna IAM.
- Kebijakan kontrol layanan (SCPs) — Tentukan izin maksimum untuk organisasi atau unit organisasi di AWS Organizations. Untuk informasi selengkapnya, lihat Kebijakan kontrol layanan dalam Panduan Pengguna AWS Organizations .
- Kebijakan kontrol sumber daya (RCPs) — Tetapkan izin maksimum yang tersedia untuk sumber daya di akun Anda. Untuk informasi selengkapnya, lihat Kebijakan kontrol sumber daya (RCPs) di Panduan AWS Organizations Pengguna.
- Kebijakan sesi – Kebijakan lanjutan yang diteruskan sebagai parameter saat membuat sesi sementara untuk peran atau pengguna terfederasi. Untuk informasi selengkapnya, lihat Kebijakan sesi dalam Panduan Pengguna IAM.

Berbagai jenis kebijakan

Ketika beberapa jenis kebijakan berlaku pada suatu permintaan, izin yang dihasilkan lebih rumit untuk dipahami. Untuk mempelajari cara AWS menentukan apakah akan mengizinkan permintaan saat beberapa jenis kebijakan terlibat, lihat Logika evaluasi kebijakan di Panduan Pengguna IAM.

Cara kerja Amazon MQ dengan IAM

Sebelum Anda menggunakan IAM untuk mengelola akses ke Amazon MQ, Anda harus memahami fitur-fitur IAM apa yang tersedia untuk digunakan dengan Amazon MQ. Untuk mendapatkan tampilan tingkat tinggi tentang cara kerja Amazon MQ dan layanan AWS lainnya dengan IAM, AWS lihat Layanan yang Bekerja dengan IAM di Panduan Pengguna IAM.

Amazon MQ menggunakan IAM untuk operasi API Amazon MQ untuk membuat, memperbarui, menghapus, dan daftar broker. Untuk akses broker untuk mempublikasikan dan berlangganan pesan, Amazon MQ untuk ActiveMQ mendukung otentikasi ActiveMQ asli dan LDAP, sementara Amazon

MQ untuk RabbitMQ mendukung otentikasi IAM dan metode lainnya. Untuk informasi selengkapnya, lihat the section called “Otentikasi dan otorisasi broker”.

Topik

- Kebijakan berbasis identitas Amazon MQ
- Kebijakan berbasis Sumber Daya Amazon MQ
- Otorisasi berbasis tanda Amazon MQ
- Peran IAM Amazon MQ

Kebijakan berbasis identitas Amazon MQ

Dengan kebijakan berbasis identitas IAM, Anda dapat menentukan tindakan dan sumber daya yang diizinkan atau ditolak, serta kondisi di mana tindakan tersebut diperbolehkan atau ditolak. Amazon MQ mendukung tindakan, sumber daya, dan kunci syarat tertentu. Untuk mempelajari semua elemen yang Anda gunakan dalam kebijakan JSON, lihat Referensi elemen kebijakan IAM JSON dalam Panduan Pengguna IAM.

Tindakan

Administrator dapat menggunakan kebijakan AWS JSON untuk menentukan siapa yang memiliki akses ke apa. Yaitu, di mana utama dapat melakukan tindakan pada sumber daya, dan dalam kondisi apa.

Elemen `Action` dari kebijakan JSON menjelaskan tindakan yang dapat Anda gunakan untuk mengizinkan atau menolak akses dalam sebuah kebijakan. Sertakan tindakan dalam kebijakan untuk memberikan izin untuk melakukan operasi terkait.

Tindakan kebijakan di Amazon MQ menggunakan prefiks berikut sebelum tindakan: `mq:`. Misalnya, untuk memberikan izin kepada seseorang untuk menjalankan instans Amazon MQ dengan operasi API `CreateBroker` Amazon MQ, Anda menyertakan tindakan `mq:CreateBroker` dalam kebijakan mereka. Pernyataan kebijakan harus memuat elemen `Action` atau `NotAction`. Amazon MQ menentukan serangkaian tindakannya sendiri yang menjelaskan tugas yang dapat Anda lakukan dengan layanan ini.

Untuk menetapkan beberapa tindakan dalam satu pernyataan, pisahkan dengan koma seperti berikut:

```
"Action": [
```

```
      "mq:action1",
      "mq:action2"
```

Anda dapat menentukan beberapa tindakan menggunakan wildcard (*). Misalnya, untuk menentukan semua tindakan yang dimulai dengan kata `Describe`, sertakan tindakan berikut:

```
"Action": "mq:Describe*"
```

Untuk melihat daftar tindakan Amazon MQ, lihat Tindakan Ditetapkan oleh Amazon MQ di Panduan Pengguna IAM.

Sumber daya

Administrator dapat menggunakan kebijakan AWS JSON untuk menentukan siapa yang memiliki akses ke apa. Yaitu, di mana utama dapat melakukan tindakan pada sumber daya, dan dalam kondisi apa.

Elemen kebijakan JSON `Resource` menentukan objek yang menjadi target penerapan tindakan. Praktik terbaiknya, tentukan sumber daya menggunakan Amazon Resource Name (ARN). Untuk tindakan yang tidak mendukung izin di tingkat sumber daya, gunakan wildcard (*) untuk menunjukkan bahwa pernyataan tersebut berlaku untuk semua sumber daya.

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

Di Amazon MQ, AWS sumber daya utama adalah broker pesan Amazon MQ dan konfigurasinya. Pialang dan konfigurasi Amazon MQ masing-masing memiliki Nama Sumber Daya Amazon (ARNs) unik yang terkait dengannya, seperti yang ditunjukkan pada tabel berikut.

Jenis Sumber Daya	ARN	Kunci kondisi
brokers	`arn:aws:mq:us-east-1:123456789012:broker:${brokerName}:${brokerId}`	aws:ResourceTag/${TagKey}
configurations	`arn:${Partition}:mq:${Region}:${Account}:configuration:${configuration-id}`	aws:ResourceTag/${TagKey}

Untuk informasi selengkapnya tentang format ARNs, lihat Amazon Resource Names (ARNs) dan Ruang Nama AWS Layanan.

Misalnya, untuk menentukan broker bernama `MyBroker` dengan BrokerId `b-1234a5b6-78cd-901e-2fgh-3i45j6k178l9` dalam pernyataan Anda, gunakan ARN berikut:

```
"Resource": "arn:aws:mq:us-
east-1:123456789012:broker:MyBroker:b-1234a5b6-78cd-901e-2fgh-3i45j6k178l9"
```

Untuk menentukan semua broker dan konfigurasi yang termasuk dalam akun tertentu, gunakan wildcard (*):

```
"Resource": "arn:aws:mq:us-east-1:123456789012:*"
```

Beberapa tindakan Amazon MQ, seperti membuat sumber daya, tidak dapat dilakukan pada sumber daya tertentu. Dalam kondisi tersebut, Anda harus menggunakan wildcard (*).

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

Tindakan API `CreateTags` memerlukan broker dan konfigurasi. Untuk menentukan beberapa sumber daya dalam satu pernyataan, pisahkan ARNs dengan koma.

```
"Resource": [
      "resource1",
      "resource2"
```

Untuk melihat daftar jenis sumber daya Amazon MQ dan jenisnya ARNs, lihat Sumber Daya yang Ditentukan oleh Amazon MQ di Panduan Pengguna IAM. Untuk mempelajari tindakan mana yang dapat menentukan ARN setiap sumber daya, lihat Tindakan yang Ditentukan oleh Amazon MQ.

Kunci syarat

Administrator dapat menggunakan kebijakan AWS JSON untuk menentukan siapa yang memiliki akses ke apa. Yaitu, principal dapat melakukan tindakan pada suatu sumber daya, dan dalam suatu syarat.

Elemen `Condition` menentukan ketika pernyataan dieksekusi berdasarkan kriteria yang ditetapkan. Anda dapat membuat ekspresi bersyarat yang menggunakan operator kondisi, misalnya sama dengan atau kurang dari, untuk mencocokkan kondisi dalam kebijakan dengan nilai-nilai yang

diminta. Untuk melihat semua kunci kondisi AWS global, lihat kunci konteks kondisi AWS global di Panduan Pengguna IAM.

Amazon MQ tidak menentukan kunci kondisi khusus layanan, tetapi mendukung penggunaan beberapa kunci syarat global. Untuk melihat daftar kunci syarat Amazon MQ, lihat tabel di bawah atau Kunci Syarat untuk Amazon MQ di Panduan Pengguna IAM. Untuk mempelajari tindakan dan sumber daya mana yang dapat Anda gunakan kunci ketentuan, lihat Tindakan yang Ditentukan oleh Amazon MQ.

Kunci kondisi	Deskripsi	Tipe
aws: RequestTag /$ {} TagKey	Filter tindakan berdasarkan tanda yang diberikan dalam permintaan.	String
aws: ResourceTag /$ {} TagKey	Filter tindakan berdasarkan tanda yang terkait dengan sumber daya.	String
aws: TagKeys	Filter tindakan berdasarkan kunci tanda yang diberikan dalam permintaan.	String

Contoh

Untuk melihat contoh identitas berbasis kebijakan Amazon MQ, lihat Contoh kebijakan berbasis Identitas Amazon MQ.

Kebijakan berbasis Sumber Daya Amazon MQ

Saat ini, Amazon MQ tidak mendukung autentikasi IAM menggunakan izin berbasis sumber daya atau kebijakan berbasis sumber daya.

Otorisasi berbasis tanda Amazon MQ

Anda dapat melampirkan tanda ke sumber daya Amazon MQ atau meneruskan tanda dalam sebuah permintaan ke Amazon MQ. Untuk mengontrol akses berdasarkan tag, Anda memberikan informasi tag di elemen kondisi kebijakan menggunakan `mq:ResourceTag/key-name`, `aws:RequestTag/key-name`, atau kunci kondisi `aws:TagKeys`.

Amazon MQ mendukung kebijakan berbasis tanda. Misalnya, Anda dapat menolak akses ke sumber daya Amazon MQ yang menyertakan tanda dengan kunci `environment` dan nilai `production`:

JSON

```
{
    "Version":"2012-10-17",
    "Statement": [
        {
            "Effect": "Deny",
            "Action": [
                "mq:DeleteBroker",
                "mq:RebootBroker",
                "mq:DeleteTags"
            ],
            "Resource": "*",
            "Condition": {
                "StringEquals": {
                    "aws:ResourceTag/environment": "production"
                }
            }
        }
    ]
}
```

Kebijakan ini akan `Deny` kemampuan untuk menghapus atau melakukan boot ulang broker Amazon MQ yang menyertakan tanda `environment/production`.

Untuk informasi selengkapnya mengenai penandaan, lihat:

- Menambahkan tag ke sumber daya Amazon MQ
- Mengontrol Akses Menggunakan Tag IAM

Peran IAM Amazon MQ

Peran IAM adalah entitas dalam AWS akun Anda yang memiliki izin tertentu.

Menggunakan kredensial sementara dengan Amazon MQ

Anda dapat menggunakan kredensial sementara untuk masuk dengan federasi, memiliki IAM role, atau menjalankan peran lintas-akun. Anda memperoleh kredensyal keamanan sementara dengan memanggil operasi AWS STS API seperti AssumeRoleatau. GetFederationToken

Amazon MQ mendukung penggunaan kredensial sementara.

Peran layanan

Fitur ini memungkinkan layanan untuk menerima peran layanan atas nama Anda. Peran ini mengizinkan layanan untuk mengakses sumber daya di layanan lain untuk menyelesaikan tindakan atas nama Anda. Peran layanan muncul di akun IAM Anda dan dimiliki oleh akun tersebut. Ini berarti administrator IAM dapat mengubah izin untuk peran ini. Namun, melakukannya mungkin merusak fungsi layanan.

Amazon MQ mendukung peran layanan.

Contoh kebijakan berbasis Identitas Amazon MQ

Secara default, pengguna dan peran tidak memiliki izin untuk membuat atau memodifikasi sumber daya Amazon MQ. Mereka juga tidak dapat melakukan tugas menggunakan Konsol Manajemen AWS, AWS CLI, atau AWS API. 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 IAM atau grup yang memerlukan izin tersebut.

Untuk mempelajari cara membuat kebijakan berbasis identitas IAM menggunakan contoh dokumen kebijakan JSON ini, lihat Membuat Kebijakan pada Tab JSON dalam Panduan Pengguna IAM.

Topik

- Praktik terbaik kebijakan
- Menggunakan konsol Amazon MQ
- Izinkan para pengguna untuk melihat izin mereka sendiri

Praktik terbaik kebijakan

Kebijakan berbasis identitas menentukan apakah seseorang dapat membuat, mengakses, atau menghapus sumber daya Amazon MQ di akun Anda. Tindakan ini membuat Akun AWS Anda dikenai biaya. Ketika Anda membuat atau mengedit kebijakan berbasis identitas, ikuti panduan dan rekomendasi ini:

- Mulailah dengan kebijakan AWS terkelola dan beralih ke izin hak istimewa paling sedikit — Untuk mulai memberikan izin kepada pengguna dan beban kerja Anda, gunakan kebijakan AWS terkelola

yang memberikan izin untuk banyak kasus penggunaan umum. Mereka tersedia di Anda Akun AWS. Kami menyarankan Anda mengurangi izin lebih lanjut dengan menentukan kebijakan yang dikelola AWS pelanggan yang khusus untuk kasus penggunaan Anda. Untuk informasi selengkapnya, lihat Kebijakan yang dikelola AWS atau Kebijakan yang dikelola AWS untuk fungsi tugas dalam Panduan Pengguna IAM.

- Menerapkan izin dengan hak akses paling rendah – Ketika Anda menetapkan izin dengan kebijakan IAM, hanya berikan izin yang diperlukan untuk melakukan tugas. Anda melakukannya dengan mendefinisikan tindakan yang dapat diambil pada sumber daya tertentu dalam kondisi tertentu, yang juga dikenal sebagai izin dengan hak akses paling rendah. Untuk informasi selengkapnya tentang cara menggunakan IAM untuk mengajukan izin, lihat Kebijakan dan izin dalam IAM dalam Panduan Pengguna IAM.
- Gunakan kondisi dalam kebijakan IAM untuk membatasi akses lebih lanjut – Anda dapat menambahkan suatu kondisi ke kebijakan Anda untuk membatasi akses ke tindakan dan sumber daya. Sebagai contoh, Anda dapat menulis kondisi kebijakan untuk menentukan bahwa semua permintaan harus dikirim menggunakan SSL. Anda juga dapat menggunakan ketentuan untuk memberikan akses ke tindakan layanan jika digunakan melalui yang spesifik Layanan AWS, seperti CloudFormation. Untuk informasi selengkapnya, lihat Elemen kebijakan JSON IAM: Kondisi dalam Panduan Pengguna IAM.
- Gunakan IAM Access Analyzer untuk memvalidasi kebijakan IAM Anda untuk memastikan izin yang aman dan fungsional – IAM Access Analyzer memvalidasi kebijakan baru dan yang sudah ada sehingga kebijakan tersebut mematuhi bahasa kebijakan IAM (JSON) dan praktik terbaik IAM. IAM Access Analyzer menyediakan lebih dari 100 pemeriksaan kebijakan dan rekomendasi yang dapat ditindaklanjuti untuk membantu Anda membuat kebijakan yang aman dan fungsional. Untuk informasi selengkapnya, lihat Validasi kebijakan dengan IAM Access Analyzer dalam Panduan Pengguna IAM.
- Memerlukan otentikasi multi-faktor (MFA) - Jika Anda memiliki skenario yang mengharuskan pengguna IAM atau pengguna root di Anda, Akun AWS aktifkan MFA untuk keamanan tambahan. Untuk meminta MFA ketika operasi API dipanggil, tambahkan kondisi MFA pada kebijakan Anda. Untuk informasi selengkapnya, lihat Amankan akses API dengan MFA dalam Panduan Pengguna IAM.

Untuk informasi selengkapnya tentang praktik terbaik dalam IAM, lihat Praktik terbaik keamanan di IAM dalam Panduan Pengguna IAM.

Menggunakan konsol Amazon MQ

Untuk mengakses konsol Amazon MQ, Anda harus memiliki satu set izin minimum. Izin ini harus memungkinkan Anda untuk membuat daftar dan melihat detail tentang sumber daya Amazon MQ di AWS akun Anda. Jika Anda membuat kebijakan berbasis identitas yang lebih ketat daripada izin minimum yang diperlukan, konsol tersebut tidak akan berfungsi sebagaimana mestinya untuk entitas (pengguna IAM atau peran) dengan kebijakan tersebut.

Untuk memastikan bahwa entitas tersebut masih dapat menggunakan konsol Amazon MQ, lampirkan juga kebijakan AWS terkelola berikut ke entitas. Untuk informasi selengkapnya, lihat Menambahkan izin ke Pengguna dalam Panduan Pengguna IAM:

```
AmazonMQReadOnlyAccess
```

Anda tidak perlu mengizinkan izin konsol minimum untuk pengguna yang melakukan panggilan hanya ke AWS CLI atau AWS API. Sebagai alternatif, hanya izinkan akses ke tindakan yang cocok dengan operasi API yang sedang Anda coba lakukan.

Izinkan para pengguna untuk melihat izin mereka sendiri

Contoh ini menunjukkan cara membuat kebijakan yang mengizinkan pengguna IAM melihat kebijakan inline dan terkelola yang dilampirkan ke identitas pengguna mereka. Kebijakan ini mencakup izin untuk menyelesaikan tindakan ini di konsol atau menggunakan API atau secara terprogram. AWS CLI AWS

```
{
    "Version": "2012-10-17",
    "Statement": [
        {
            "Sid": "ViewOwnUserInfo",
            "Effect": "Allow",
            "Action": [
                "iam:GetUserPolicy",
                "iam:ListGroupsForUser",
                "iam:ListAttachedUserPolicies",
                "iam:ListUserPolicies",
                "iam:GetUser"
            ],
            "Resource": ["arn:aws:iam::*:user/${aws:username}"]
        },
```

```
        {
            "Sid": "NavigateInConsole",
            "Effect": "Allow",
            "Action": [
                "iam:GetGroupPolicy",
                "iam:GetPolicyVersion",
                "iam:GetPolicy",
                "iam:ListAttachedGroupPolicies",
                "iam:ListGroupPolicies",
                "iam:ListPolicyVersions",
                "iam:ListPolicies",
                "iam:ListUsers"
            ],
            "Resource": "*"
        }
    ]
}
```

Autentikasi dan otorisasi API untuk Amazon MQ

Amazon MQ menggunakan penandatanganan AWS permintaan standar untuk otentikasi API. Untuk informasi selengkapnya, lihat Menandatangani Permintaan API AWS di Referensi Umum AWS.

> **Note**
>
> Saat ini, Amazon MQ tidak mendukung autentikasi IAM menggunakan izin berbasis sumber daya atau kebijakan berbasis sumber daya.

Untuk mengizinkan AWS pengguna untuk bekerja dengan broker, konfigurasi, dan pengguna, Anda harus mengedit izin kebijakan IAM Anda.

Topik

- Izin IAM yang Diperlukan untuk Membuat Broker Amazon MQ
- Referensi izin REST API Amazon MQ
- Referensi izin tambahan Amazon MQ
- Izin tingkat sumber daya untuk tindakan API Amazon MQ

Izin IAM yang Diperlukan untuk Membuat Broker Amazon MQ

Untuk membuat broker, Anda harus menggunakan kebijakan IAM `AmazonMQFullAccess` atau termasuk izin EC2 berikut dalam kebijakan IAM.

Kebijakan kustom berikut ini terdiri dari dua pernyataan (satu bersyarat) yang memberikan izin untuk memanipulasi sumber daya yang diperlukan Amazon MQ untuk membuat broker ActiveMQ.

> **Important**
>
> - Tindakan `ec2:CreateNetworkInterface` diperlukan untuk mengizinkan Amazon MQ membuat antarmuka jaringan elastis (ENI) di akun Anda atas nama Anda.
> - Tindakan `ec2:CreateNetworkInterfacePermission` mengotorisasi Amazon MQ untuk melampirkan ENI ke broker ActiveMQ.
> - Kunci syarat `ec2:AuthorizedService` memastikan bahwa izin ENI dapat diberikan hanya untuk akun layanan Amazon MQ.

JSON

```
{
    "Version":"2012-10-17",
    "Statement": [{
        "Action": [
            "mq:*",
            "ec2:CreateNetworkInterface",
            "ec2:DeleteNetworkInterface",
            "ec2:DetachNetworkInterface",
            "ec2:DescribeInternetGateways",
            "ec2:DescribeNetworkInterfaces",
            "ec2:DescribeRouteTables",
            "ec2:DescribeSecurityGroups",
            "ec2:DescribeSubnets",
            "ec2:DescribeVpcs"
        ],
        "Effect": "Allow",
        "Resource": "*"
    },{
        "Action": [
            "ec2:CreateNetworkInterfacePermission",
```

```
            "ec2:DeleteNetworkInterfacePermission",
            "ec2:DescribeNetworkInterfacePermissions"
        ],
        "Effect": "Allow",
        "Resource": "*",
        "Condition": {
            "StringEquals": {
                "ec2:AuthorizedService": "mq.amazonaws.com"
            }
        }
    }]
}
```

Untuk informasi lebih lanjut, lihat Langkah 2: buat pengguna dan dapatkan AWS kredensialnya dan Jangan Pernah Memodifikasi atau Menghapus Antarmuka Jaringan Elastis Amazon MQ.

Referensi izin REST API Amazon MQ

Tabel berikut mencantumkan Amazon MQ REST APIs dan izin IAM yang sesuai.

Amazon MQ REST APIs dan Izin yang Diperlukan

Amazon MQ REST APIs	Izin yang Diperlukan
CreateBroker	mq:CreateBroker
CreateConfiguration	mq:CreateConfiguration
CreateTags	mq:CreateTags
CreateUser	mq:CreateUser
DeleteBroker	mq:DeleteBroker
DeleteUser	mq:DeleteUser
DescribeBroker	mq:DescribeBroker
DescribeConfiguration	mq:DescribeConfiguration
DescribeConfigurationRevision	mq:DescribeConfigurationRevision

Amazon MQ REST APIs	Izin yang Diperlukan
DescribeUser	mq:DescribeUser
ListBrokers	mq:ListBrokers
ListConfigurationRevisions	mq:ListConfigurationRevisions
ListConfigurations	mq:ListConfigurations
ListTags	mq:ListTags
ListUsers	mq:ListUsers
RebootBroker	mq:RebootBroker
UpdateBroker	mq:UpdateBroker
UpdateConfiguration	mq:UpdateConfiguration
UpdateUser	mq:UpdateUser

Referensi izin tambahan Amazon MQ

Tabel berikut mencantumkan API Amazon MQ dan izin IAM tambahan yang diperlukan untuk fitur tertentu, seperti OAuth otentikasi 2.0.

API REST Amazon MQ	Izin	Deskripsi
UpdateBroker	mq:UpdateBrokerAccessConfiguration	Anda memerlukan izin ini untuk memperbarui opsi otentikasi dan otorisasi dalam konfigurasi broker terkait. Untuk informasi selengkapnya, lihat OAuth 2.0 otentikasi dan otorisasi untuk Amazon MQ untuk RabbitMQ.

Izin tingkat sumber daya untuk tindakan API Amazon MQ

Istilah izin tingkat sumber daya mengacu pada kemampuan untuk menentukan sumber daya yang boleh digunakan pengguna untuk melakukan tindakan. Amazon MQ memiliki dukungan parsial untuk izin tingkat sumber daya. Untuk tindakan Amazon MQ tertentu, Anda dapat mengontrol kapan pengguna diizinkan untuk menggunakan tindakan tersebut berdasarkan ketentuan yang harus dipenuhi, atau sumber daya tertentu yang diizinkan untuk digunakan oleh pengguna.

Tabel berikut menjelaskan tindakan Amazon MQ API yang saat ini mendukung izin tingkat sumber daya, serta kunci sumber daya, sumber daya, dan kondisi yang didukung untuk setiap tindakan. ARNs

> **Important**
>
> Jika tindakan API Amazon MQ tidak tercantum dalam tabel ini, artinya izin tingkat sumber daya tidak didukung. Jika tindakan API Amazon MQ tidak mendukung izin tingkat sumber daya, Anda dapat memberikan pengguna izin untuk menggunakan tindakan, tetapi Anda harus menentukan wildcard * untuk elemen sumber daya pernyataan kebijakan.

Tindakan API	Jenis Sumber Daya (*wajib)
CreateConfiguration	konfigurasi*
CreateTags	broker, konfigurasi
CreateUser	pialang*
DeleteBroker	pialang*
DeleteUser	pialang*
DescribeBroker	pialang*
DescribeConfiguration	konfigurasi*
DescribeConfigurationRevision	konfigurasi*
DescribeUser	pialang*

Tindakan API	Jenis Sumber Daya (*wajib)
ListConfigurationRevisions	konfigurasi*
ListConfigurationRevisions	konfigurasi*
ListTags	broker, konfigurasi
ListUsers	pialang*
RebootBroker	pialang*
UpdateBroker	pialang*
UpdateConfiguration	konfigurasi*
UpdateUser	pialang*

Otentikasi dan otorisasi broker

Amazon MQ menyediakan metode otentikasi dan otorisasi yang berbeda tergantung pada jenis mesin broker Anda.

Otentikasi dan otorisasi untuk Amazon MQ untuk ActiveMQ

Amazon MQ untuk ActiveMQ mendukung metode otentikasi dan otorisasi berikut:

Otentikasi dan otorisasi sederhana

Dalam metode ini, pengguna broker dibuat dan dikelola melalui konsol Amazon MQ atau API. Pengguna dapat dikonfigurasi dengan izin khusus untuk mengakses antrian, topik, dan ActiveMQ Web Console. Untuk informasi selengkapnya tentang metode ini, lihat Membuat pengguna broker ActiveMQ.

Otentikasi dan otorisasi LDAP

Dalam metode ini, pengguna broker mengautentikasi melalui kredensyal yang disimpan di server LDAP Anda. Anda dapat menambahkan, menghapus, dan memodifikasi pengguna dan menetapkan

izin untuk topik dan antrian melalui server LDAP, menyediakan otentikasi dan otorisasi terpusat. Untuk informasi lebih lanjut tentang metode ini, lihat Mengintegrasikan broker ActiveMQ dengan LDAP.

Otentikasi dan otorisasi untuk Amazon MQ untuk RabbitMQ

Amazon MQ untuk RabbitMQ mendukung metode otentikasi dan otorisasi berikut:

Otentikasi dan otorisasi sederhana

Dalam metode ini, pengguna broker disimpan secara internal di broker RabbitMQ dan dikelola melalui konsol web atau API manajemen. Izin untuk vhost, pertukaran, antrian, dan topik dikonfigurasi langsung di RabbitMQ. Ini adalah metode default. Untuk informasi selengkapnya, lihat Otentikasi dan otorisasi sederhana.

OAuth 2.0 otentikasi dan otorisasi

Dalam metode ini, pengguna broker dan izin mereka dikelola oleh penyedia identitas OAuth 2.0 eksternal (iDP). Otentikasi pengguna dan izin sumber daya untuk vhost, pertukaran, antrian, dan topik dipusatkan melalui sistem lingkup penyedia 2.0. OAuth Ini menyederhanakan manajemen pengguna dan memungkinkan integrasi dengan sistem identitas yang ada. Untuk informasi selengkapnya, lihat otentikasi dan otorisasi OAuth 2.0.

Otentikasi dan otorisasi IAM

Dalam metode ini, pengguna broker mengautentikasi menggunakan kredensyal AWS IAM melalui federasi keluar IAM. Kredensi IAM digunakan untuk mendapatkan token JWT dari AWS Security Token Service (STS), dan token JWT ini berfungsi sebagai token 2.0 untuk otentikasi. OAuth Metode ini memanfaatkan dukungan OAuth 2.0 yang ada di Amazon MQ untuk RabbitMQ, AWS di mana bertindak sebagai penyedia identitas 2.0. OAuth Otentikasi pengguna ditangani oleh AWS IAM, sementara izin sumber daya untuk vhost, pertukaran, antrian, dan topik dikelola melalui kebijakan IAM dan alias ruang lingkup yang dikonfigurasi di RabbitMQ. Untuk informasi selengkapnya, lihat autentikasi dan otorisasi IAM.

Otentikasi dan otorisasi LDAP

Dalam metode ini, pengguna broker dan izin mereka dikelola oleh layanan direktori LDAP eksternal. Otentikasi pengguna dan izin sumber daya dipusatkan melalui server LDAP, memungkinkan pengguna untuk mengakses RabbitMQ menggunakan kredensi layanan direktori yang ada. Untuk informasi selengkapnya, lihat otentikasi dan otorisasi LDAP.

Otentikasi dan otorisasi HTTP

Dalam metode ini, pengguna broker dan izin mereka dikelola oleh server HTTP eksternal. Otentikasi pengguna dan izin sumber daya dipusatkan melalui server HTTP, memungkinkan pengguna untuk mengakses RabbitMQ menggunakan penyedia Otentikasi dan Otorisasi mereka sendiri. Untuk informasi selengkapnya tentang metode ini, lihat otentikasi dan otorisasi HTTP.

Otentikasi sertifikat SSL

Amazon MQ mendukung TLS bersama (MTLS) untuk broker RabbitMQ. Plugin otentikasi SSL menggunakan sertifikat klien dari koneksi mTLS untuk mengautentikasi pengguna. Dalam metode ini, pengguna broker diautentikasi menggunakan sertifikat klien X.509 alih-alih kredensyal nama pengguna dan kata sandi. Sertifikat klien divalidasi terhadap Otoritas Sertifikat (CA) tepercaya, dan nama pengguna diekstraksi dari bidang dalam sertifikat, seperti Nama Umum (CN) atau Nama Alternatif Subjek (SAN). Metode ini memberikan otentikasi yang kuat tanpa mentransmisikan kredensyal melalui jaringan. Untuk informasi selengkapnya, lihat otentikasi sertifikat SSL.

> **Note**
>
> RabbitMQ mendukung beberapa metode otentikasi dan otorisasi untuk digunakan secara bersamaan. Misalnya, Anda dapat mengaktifkan otentikasi OAuth 2.0 dan sederhana (internal). Untuk informasi lebih lanjut, lihat bagian tutorial OAuth 2.0 tentang mengaktifkan otentikasi OAuth 2.0 dan sederhana (internal) dan dokumentasi kontrol akses RabbitMQ. Amazon MQ merekomendasikan untuk membuat pengguna internal saat menguji konfigurasi otentikasi. Hal ini memungkinkan konfigurasi akses untuk divalidasi menggunakan RabbitMQ management API. Untuk informasi selengkapnya, lihat Validasi akses.

AWS kebijakan terkelola untuk Amazon MQ

Kebijakan AWS terkelola adalah kebijakan mandiri yang dibuat dan dikelola oleh AWS. AWS Kebijakan terkelola dirancang untuk memberikan izin bagi banyak kasus penggunaan umum sehingga Anda dapat mulai menetapkan izin kepada pengguna, grup, dan peran.

Perlu diingat bahwa kebijakan AWS terkelola mungkin tidak memberikan izin hak istimewa paling sedikit untuk kasus penggunaan spesifik Anda karena tersedia untuk digunakan semua pelanggan. AWS Kami menyarankan Anda untuk mengurangi izin lebih lanjut dengan menentukan kebijakan yang dikelola pelanggan yang khusus untuk kasus penggunaan Anda.

Anda tidak dapat mengubah izin yang ditentukan dalam kebijakan AWS terkelola. Jika AWS memperbarui izin yang ditentukan dalam kebijakan AWS terkelola, pembaruan akan memengaruhi semua identitas utama (pengguna, grup, dan peran) yang dilampirkan kebijakan tersebut. AWS kemungkinan besar akan memperbarui kebijakan AWS terkelola saat baru Layanan AWS diluncurkan atau operasi API baru tersedia untuk layanan yang ada.

Untuk informasi selengkapnya, lihat Kebijakan terkelola AWS dalam Panduan Pengguna IAM.

Amazon MQ mendukung kebijakan AWS terkelola berikut:

- Amazon MQApi FullAccess
- Amazon MQApi ReadOnlyAccess
- MQFullAkses Amazon
- Amazon MQRead OnlyAccess
- Amazon MQService RolePolicy

AWS kebijakan terkelola: Amazon MQService RolePolicy

Anda tidak dapat melampirkan `AmazonMQServiceRolePolicy` ke entitas IAM Anda. Kebijakan ini dilampirkan ke peran tertaut layanan yang mengizinkan Amazon MQ untuk melakukan tindakan atas nama Anda. Untuk informasi selengkapnya tentang kebijakan izin ini dan tindakan yang mengizinkan Amazon MQ untuk bekerja, lihat the section called “Izin peran tertaut layanan untuk Amazon MQ”.

Pembaruan Amazon MQ ke AWS kebijakan terkelola

Lihat detail tentang pembaruan kebijakan AWS terkelola untuk Amazon MQ sejak layanan ini mulai melacak perubahan ini. Untuk pemberitahuan otomatis tentang perubahan halaman ini, berlangganan ke umpan RSS pada halaman riwayat Dokumen Amazon MQ.

Perubahan	Deskripsi	Tanggal
Amazon MQ mulai melacak perubahan	Amazon MQ mulai melacak perubahan untuk kebijakan yang AWS dikelola.	5 Mei 2021

Menggunakan peran tertaut layanan untuk Amazon MQ

Amazon MQ menggunakan peran terkait layanan AWS Identity and Access Management (IAM). Peran tertaut layanan adalah tipe IAM role unik yang ditautkan langsung ke Amazon MQ. Peran terkait layanan telah ditentukan sebelumnya oleh Amazon MQ dan mencakup semua izin yang diperlukan layanan untuk memanggil layanan lain atas nama Anda. AWS

Peran tertaut layanan mempermudah pengaturan Amazon MQ karena Anda tidak perlu menambahkan izin yang diperlukan secara manual. Amazon MQ menentukan izin dari peran tertaut layanan, kecuali jika ditentukan lain, hanya Amazon MQ yang dapat mengambil perannya. Izin yang ditentukan mencakup kebijakan kepercayaan dan kebijakan izin, serta bahwa kebijakan izin tidak dapat dilampirkan ke entitas IAM lainnya.

Anda dapat menghapus peran yang terhubung dengan layanan hanya setelah menghapus sumber daya terkait terlebih dahulu. Ini dapat melindungi sumber daya Amazon MQ karena Anda tidak dapat secara ceroboh menghapus izin untuk mengakses sumber daya.

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. Pilih Ya dengan tautan untuk melihat dokumentasi peran terkait layanan untuk layanan tersebut.

Izin peran tertaut layanan untuk Amazon MQ

Amazon MQ menggunakan peran terkait layanan bernama MQ AWSServiceRoleForAmazon—Amazon MQ menggunakan peran terkait layanan ini untuk memanggil layanan atas nama Anda. AWS

Peran terkait layanan AWSService RoleForAmazon MQ mempercayai layanan berikut untuk mengambil peran:

- `mq.amazonaws.com`

Amazon MQ menggunakan kebijakan izin `AmazonMQServiceRolePolicy`, yang dilampirkan ke peran terkait layanan AWSService RoleForAmazon MQ, untuk menyelesaikan tindakan berikut pada sumber daya yang ditentukan:

- Tindakan: `ec2:CreateVpcEndpoint` pada sumber daya `vpc`.
- Tindakan: `ec2:CreateVpcEndpoint` pada sumber daya `subnet`.

- Tindakan: `ec2:CreateVpcEndpoint` pada sumber daya `security-group`.
- Tindakan: `ec2:CreateVpcEndpoint` pada sumber daya `vpc-endpoint`.
- Tindakan: `ec2:DescribeVpcEndpoints` pada sumber daya `vpc`.
- Tindakan: `ec2:DescribeVpcEndpoints` pada sumber daya `subnet`.
- Tindakan: `ec2:CreateTags` pada sumber daya `vpc-endpoint`.
- Tindakan: `logs:PutLogEvents` pada sumber daya `log-group`.
- Tindakan: `logs:DescribeLogStreams` pada sumber daya `log-group`.
- Tindakan: `logs:DescribeLogGroups` pada sumber daya `log-group`.
- Tindakan: `CreateLogStream` pada sumber daya `log-group`.
- Tindakan: `CreateLogGroup` pada sumber daya `log-group`.

Saat Anda membuat broker Amazon MQ for RabbitMQ, kebijakan izin `AmazonMQServiceRolePolicy` memungkinkan Amazon MQ melakukan tugas berikut atas nama Anda.

- Membuat VPC endpoint untuk broker menggunakan Amazon VPC, subnet, dan grup keamanan yang Anda berikan. Anda dapat menggunakan titik akhir yang dibuat untuk broker agar terhubung ke broker melalui konsol manajemen RabbitMQ, API manajemen, atau secara pemrograman.
- Buat grup log, dan publikasikan log broker ke Amazon CloudWatch Logs.

JSON

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

```
                "ec2:DescribeVpcEndpoints"
            ],
            "Resource": "*"
        },
        {
            "Effect": "Allow",
            "Action": [
                "ec2:CreateVpcEndpoint"
            ],
            "Resource": [
                "arn:aws:ec2:*:*:vpc/*",
                "arn:aws:ec2:*:*:subnet/*",
                "arn:aws:ec2:*:*:security-group/*"
            ]
        },
        {
            "Effect": "Allow",
            "Action": [
                "ec2:CreateVpcEndpoint"
            ],
            "Resource": [
                "arn:aws:ec2:*:*:vpc-endpoint/*"
            ],
            "Condition": {
                "StringEquals": {
                    "aws:RequestTag/AMQManaged": "true"
                }
            }
        },
        {
            "Effect": "Allow",
            "Action": [
                "ec2:CreateTags"
            ],
            "Resource": "arn:aws:ec2:*:*:vpc-endpoint/*",
            "Condition": {
                "StringEquals": {
                    "ec2:CreateAction": "CreateVpcEndpoint"
                }
            }
        },
        {
            "Effect": "Allow",
            "Action": [
```

```
                "ec2:DeleteVpcEndpoints"
            ],
            "Resource": "arn:aws:ec2:*:*:vpc-endpoint/*",
            "Condition": {
                "StringEquals": {
                    "ec2:ResourceTag/AMQManaged": "true"
                }
            }
        },
        {
            "Effect": "Allow",
            "Action": [
                "logs:PutLogEvents",
                "logs:DescribeLogStreams",
                "logs:DescribeLogGroups",
                "logs:CreateLogStream",
                "logs:CreateLogGroup"
            ],
            "Resource": [
                "arn:aws:logs:*:*:log-group:/aws/amazonmq/*"
            ]
        }
    ]
}
```

Anda harus mengonfigurasi izin untuk mengizinkan entitas IAM (seperti pengguna, grup, atau peran) untuk membuat, mengedit, atau menghapus peran terkait layanan. Untuk informasi lebih lanjut, lihat Izin Peran yang Terhubung dengan Layanan di Panduan Pengguna IAM.

Membuat peran tertaut layanan untuk Amazon MQ

Anda tidak perlu membuat peran terkait layanan secara manual. Saat pertama kali membuat broker, Amazon MQ menciptakan peran terkait layanan untuk memanggil AWS layanan atas nama Anda. Semua broker berikutnya yang Anda buat akan menggunakan peran yang sama dan tidak ada peran baru yang dibuat.

> **Important**
>
> Peran tertaut layanan ini dapat muncul di akun Anda jika Anda telah menyelesaikan tindakan di layanan lain yang menggunakan fitur yang didukung oleh peran ini. Untuk mempelajari lebih lanjut, lihat Peran Baru yang Muncul di Akun IAM Saya.

Jika Anda menghapus peran terkait layanan ini, lalu ingin membuatnya lagi, Anda dapat menggunakan proses yang sama untuk membuat ulang peran tersebut di akun Anda.

Anda juga dapat menggunakan konsol IAM untuk membuat peran terkait layanan dengan kasus penggunaan Amazon MQ. Di AWS CLI atau AWS API, buat peran terkait layanan dengan nama `mq.amazonaws.com` layanan. Untuk informasi selengkapnya, lihat Membuat peran tertaut layanan dalam Panduan Pengguna IAM. Jika Anda menghapus peran tertaut layanan ini, Anda dapat mengulang proses yang sama untuk membuat peran tersebut lagi.

> **Important**
>
> Peran Tertaut Layanan hanya dibuat untuk Amazon MQ untuk RabbitMQ.

Mengedit peran tertaut layanan untuk Amazon MQ

Amazon MQ tidak mengizinkan Anda mengedit peran terkait layanan AWSService RoleForAmazon MQ. Namun, Anda dapat mengedit penjelasan peran menggunakan IAM. Untuk informasi lebih lanjut, lihat Mengedit peran terkait layanan dalam Panduan Pengguna IAM.

Menghapus peran tertaut layanan untuk Amazon MQ

Jika Anda tidak perlu lagi menggunakan fitur atau layanan yang memerlukan peran terkait layanan, kami menyarankan Anda menghapus peran tersebut. Dengan begitu, Anda tidak memiliki entitas yang tidak digunakan yang tidak dipantau atau dipelihara secara aktif. Tetapi, Anda harus membersihkan sumber daya peran yang terhubung dengan layanan sebelum menghapusnya secara manual.

> **Note**
>
> Jika layanan Amazon MQ menggunakan peran saat Anda mencoba untuk menghapus sumber daya, maka penghapusan tersebut kemungkinan gagal. Jika hal itu terjadi, tunggu beberapa menit dan coba mengoperasikannya lagi.

Untuk menghapus sumber daya Amazon MQ yang digunakan oleh MQ AWSService RoleForAmazon

- Hapus broker Amazon MQ Anda menggunakan, Amazon MQ CLI Konsol Manajemen AWS, atau Amazon MQ API. Untuk informasi lebih lanjut tentang penghapusan broker, lihat ???.

Untuk menghapus peran tertaut layanan secara manual menggunakan IAM

Gunakan konsol IAM, the AWS CLI, atau AWS API untuk menghapus peran terkait layanan AWSService RoleForAmazon MQ. Untuk informasi lebih lanjut, lihat Menghapus Peran Tertaut Layanan dalam Panduan Pengguna IAM.

Wilayah yang didukung untuk peran terkait layanan Amazon MQ

Amazon MQ mendukung penggunaan peran terkait layanan di semua wilayah tempat layanan tersedia. Untuk informasi lebih lanjut, lihat Wilayah dan titik akhir AWS.

Nama wilayah	Identitas wilayah	Dukungan di Amazon MQ
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

Nama wilayah	Identitas wilayah	Dukungan di Amazon MQ
Asia Pacific (Sydney)	ap-southeast-2	Ya
Asia Pacific (Tokyo)	ap-northeast-1	Ya
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
AWS GovCloud (US)	us-gov-west-1	Tidak

Pemecahan masalah identitas dan akses Amazon MQ

Gunakan informasi berikut untuk membantu Anda mendiagnosis dan mengatasi masalah umum yang mungkin Anda temui saat bekerja menggunakan Amazon MQ dan IAM.

Topik

- Saya tidak Berwenang untuk Melakukan Tindakan di Amazon MQ
- Saya tidak berwenang untuk melakukan iam: PassRole
- Saya ingin mengizinkan orang di luar AWS akun saya untuk mengakses sumber daya Amazon MQ saya

Saya tidak Berwenang untuk Melakukan Tindakan di Amazon MQ

Jika Konsol Manajemen AWS memberitahu Anda bahwa Anda tidak berwenang untuk melakukan tindakan, maka Anda harus menghubungi administrator Anda untuk bantuan. Administrator Anda adalah orang yang memberi Anda kredensial masuk.

Contoh kesalahan berikut terjadi ketika `mateojackson` pengguna mencoba menggunakan konsol untuk melihat detail tentang *`widget`* tetapi tidak memiliki `mq:GetWidget` izin.

```
User: arn:aws:iam::123456789012:user/mateojackson is not authorized to perform:
 mq:GetWidget on resource: my-example-widget
```

Dalam hal ini, Mateo meminta administratornya untuk memperbarui kebijakannya untuk mengizinkan dia mengakses sumber daya *`my-example-widget`* menggunakan tindakan `mq: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 MQ.

Beberapa Layanan AWS memungkinkan Anda untuk meneruskan peran yang ada ke layanan tersebut alih-alih membuat peran layanan baru atau peran terkait layanan. Untuk melakukannya, Anda harus memiliki izin untuk meneruskan peran ke layanan.

Contoh kesalahan berikut terjadi saat pengguna IAM bernama `marymajor` mencoba menggunakan konsol tersebut untuk melakukan tindakan di Amazon MQ. 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 memerlukan bantuan, hubungi AWS administrator Anda. Administrator Anda adalah orang yang memberi Anda kredensial masuk.

Saya ingin mengizinkan orang di luar AWS akun saya untuk mengakses sumber daya Amazon MQ saya

Anda dapat membuat peran yang dapat digunakan pengguna di akun lain atau orang-orang di luar organisasi Anda untuk mengakses sumber daya Anda. Anda dapat menentukan siapa saja yang dipercaya untuk mengambil peran tersebut. Untuk layanan yang mendukung kebijakan berbasis sumber daya atau daftar kontrol akses (ACLs), Anda dapat menggunakan kebijakan tersebut untuk memberi orang akses ke sumber daya Anda.

Untuk mempelajari selengkapnya, periksa referensi berikut:

- Untuk mempelajari apakah Amazon MQ mendukung fitur ini, lihat Cara kerja Amazon MQ dengan IAM.
- Untuk mempelajari cara menyediakan akses ke sumber daya Anda di seluruh sumber daya Akun AWS yang Anda miliki, lihat Menyediakan akses ke pengguna IAM di pengguna lain Akun AWS yang Anda miliki di Panduan Pengguna IAM.
- Untuk mempelajari cara menyediakan akses ke sumber daya Anda kepada pihak ketiga Akun AWS, lihat Menyediakan akses yang Akun AWS dimiliki oleh pihak ketiga dalam Panduan Pengguna IAM.
- Untuk mempelajari cara memberikan akses melalui federasi identitas, lihat Menyediakan akses ke pengguna terautentikasi eksternal (federasi identitas) dalam Panduan Pengguna IAM.
- Untuk mempelajari perbedaan antara menggunakan peran dan kebijakan berbasis sumber daya untuk akses lintas akun, lihat Akses sumber daya lintas akun di IAM di Panduan Pengguna IAM.

Validasi kepatuhan untuk Amazon MQ

Auditor pihak ketiga menilai keamanan dan kepatuhan Amazon MQ sebagai bagian dari AWS beberapa program kepatuhan. Hal ini mencakup SOC, PCI, HIPAA, dan lainnya.

Untuk mempelajari apakah an Layanan AWS berada dalam lingkup program kepatuhan tertentu, lihat Layanan AWS di Lingkup oleh Program Kepatuhan Layanan AWS dan pilih program kepatuhan yang Anda minati. Untuk informasi umum, lihat Program AWS Kepatuhan Program AWS .

Anda dapat mengunduh laporan audit pihak ketiga menggunakan AWS Artifact. Untuk informasi selengkapnya, lihat Mengunduh Laporan di AWS Artifact .

Tanggung jawab kepatuhan Anda saat menggunakan Layanan AWS ditentukan oleh sensitivitas data Anda, tujuan kepatuhan perusahaan Anda, dan hukum dan peraturan yang berlaku. Untuk informasi selengkapnya tentang tanggung jawab kepatuhan Anda saat menggunakan Layanan AWS, lihat Dokumentasi AWS Keamanan.

Ketahanan di Amazon MQ

Infrastruktur AWS global dibangun di sekitar AWS Wilayah dan Zona Ketersediaan. AWS Wilayah menyediakan beberapa Availability Zone yang terpisah secara fisik dan terisolasi, yang terhubung

dengan latensi rendah, throughput tinggi, dan jaringan yang sangat redundan. Dengan Availability Zone, Anda dapat mendesain dan mengoperasikan aplikasi dan basis data yang secara otomatis mengalami kegagalan di antara zona tanpa gangguan. Zona Ketersediaan memiliki ketersediaan dan toleransi kesalahan yang lebih baik, dan dapat diskalakan dibandingkan infrastruktur pusat data tunggal atau multi tradisional.

Untuk informasi selengkapnya tentang AWS Wilayah dan Availability Zone, lihat Infrastruktur AWS Global.

Keamanan infrastruktur di Amazon MQ

Sebagai layanan terkelola, Amazon MQ dilindungi oleh keamanan jaringan AWS global. Untuk informasi tentang layanan AWS keamanan dan cara AWS melindungi infrastruktur, lihat Keamanan AWS Cloud. Untuk mendesain AWS lingkungan Anda menggunakan praktik terbaik untuk keamanan infrastruktur, lihat Perlindungan Infrastruktur dalam Kerangka Kerja yang AWS Diarsiteksikan dengan Baik Pilar Keamanan.

Anda menggunakan panggilan API yang AWS dipublikasikan untuk mengakses Amazon MQ melalui jaringan. Klien harus mendukung hal-hal berikut:

- Keamanan Lapisan Pengangkutan (TLS). Kami mensyaratkan TLS 1.2 dan menganjurkan TLS 1.3.
- Sandi cocok dengan sistem kerahasiaan maju sempurna (perfect forward secrecy, PFS) seperti DHE (Ephemeral Diffie-Hellman) atau ECDHE (Elliptic Curve Ephemeral Diffie-Hellman). Sebagian besar sistem modern seperti Java 7 dan versi lebih baru mendukung mode-mode ini.

Praktik terbaik keamanan untuk Amazon MQ

Pola desain berikut dapat meningkatkan keamanan broker Amazon MQ Anda.

Topik

- Lebih memilih broker tanpa aksesibilitas publik
- Selalu konfigurasikan peta otorisasi
- Blokir protokol yang tidak diperlukan dengan grup keamanan VPC

Untuk informasi selengkapnya tentang cara Amazon MQ mengenkripsi data Anda, serta daftar protokol yang didukung, lihat Perlindungan Data.

Lebih memilih broker tanpa aksesibilitas publik

Broker yang dibuat tanpa aksesibilitas publik tidak dapat diakses dari luar VPC Anda. Ini sangat mengurangi kerentanan broker Anda terhadap serangan Distributed Denial of Service (DDoS) dari internet publik. Untuk informasi selengkapnya, lihat Cara Membantu Mempersiapkan Serangan DDoS dengan Mengurangi Permukaan Serangan Anda di Blog AWS Keamanan.

Selalu konfigurasikan peta otorisasi

Karena ActiveMQ tidak memiliki peta otorisasi yang dikonfigurasi secara default, setiap pengguna yang diautentikasi dapat melakukan tindakan apa pun pada broker. Dengan demikian, praktik terbaiknya adalah membatasi izin menurut grup. Untuk informasi selengkapnya, lihat `authorizationEntry`.

> **Important**
>
> Jika menentukan peta otorisasi yang tidak menyertakan grup `activemq-webconsole`, Anda tidak dapat menggunakan Konsol Web ActiveMQ karena grup tidak berwenang untuk mengirim pesan ke, atau menerima pesan dari, broker Amazon MQ.

Blokir protokol yang tidak diperlukan dengan grup keamanan VPC

Untuk meningkatkan keamanan bagi broker swasta, Anda harus membatasi koneksi protokol dan port yang tidak perlu dengan mengonfigurasi Grup Keamanan VPC Amazon Anda dengan benar. Misalnya, untuk membatasi akses ke sebagian besar protokol sambil mengizinkan akses ke OpenWire dan konsol web, Anda dapat mengizinkan akses ke hanya 61617 dan 8162. Ini membatasi eksposur Anda dengan memblokir protokol yang tidak Anda gunakan, sambil mengizinkan OpenWire dan konsol web berfungsi secara normal.

Hanya memungkinkan port protokol yang Anda gunakan.

- AMQP: 5671
- MQTT: 8883
- OpenWire: 61617
- STOMP: 61614
- WebSocket: 61619

Untuk informasi selengkapnya, lihat:

- Grup Keamanan untuk VPC Anda
- Grup Keamanan Default untuk VPC Anda
- Bekerja dengan Kelompok Keamanan

Pencatatan dan pemantauan broker Amazon MQ

Pemantauan adalah bagian penting dalam menjaga keandalan, ketersediaan, dan kinerja AWS solusi Anda. Anda harus mengumpulkan data pemantauan dari semua bagian AWS solusi Anda sehingga Anda dapat lebih mudah men-debug kegagalan multi-titik jika terjadi. AWS menyediakan beberapa alat untuk memantau sumber daya Amazon MQ Anda dan menanggapi potensi insiden:

Anda dapat menggunakan CloudWatch untuk melihat dan menganalisis metrik untuk broker Amazon MQ Anda. Anda dapat melihat dan menganalisis metrik broker Anda dari CloudWatch konsol, konsol AWS CLI, atau. CloudWatch AWS CLI CloudWatch metrik untuk Amazon MQ secara otomatis disurvei dari broker dan kemudian didorong ke CloudWatch setiap menit. Untuk broker ActiveMQ CloudWatch , monitor hanya 1000 tujuan pertama.. Untuk broker RabbitMQ, hanya CloudWatch memantau 500 tujuan pertama, dipesan berdasarkan jumlah konsumen..

Untuk daftar lengkap metrik Amazon MQ, lihat CloudWatch Metrik yang tersedia Amazon MQ untuk broker ActiveMQ.

Untuk informasi tentang membuat CloudWatch alarm untuk metrik, lihat Membuat atau Mengedit CloudWatch Alarm di Panduan CloudWatch Pengguna Amazon.

Mengakses CloudWatch metrik untuk Amazon MQ

Anda dapat mengakses CloudWatch metrik menggunakan Konsol Manajemen AWS, AWS CLI, dan API.

Anda mungkin ingin mengakses CloudWatch metrik tanpa menggunakan. Konsol Manajemen AWS

Untuk mengakses metrik Amazon MQ menggunakan AWS CLI, gunakan perintah. `get-metric-statistics` Untuk informasi selengkapnya, lihat Mendapatkan Statistik untuk Metrik di Panduan CloudWatch Pengguna Amazon.

Untuk mengakses metrik Amazon MQ menggunakan CloudWatch API, gunakan tindakan. `GetMetricStatistics` Untuk informasi selengkapnya, lihat Mendapatkan Statistik untuk Metrik di Panduan CloudWatch Pengguna Amazon.

Mengakses CloudWatch metrik menggunakan Konsol Manajemen AWS

Contoh berikut menunjukkan cara mengakses CloudWatch metrik untuk Amazon MQ menggunakan .Jika Anda sudah masuk Konsol Manajemen AWS ke konsol Amazon MQ, pada halaman Detail broker, pilih Tindakan, Lihat metrik. CloudWatch

1. Masuk ke konsol CloudWatch tersebut.
2. Di panel navigasi, pilih Metrics (Metrik).
3. Pilih namespace metrik AmazonMQ.
4. Pilih salah satu dimensi metrik berikut:

   - Metrik Pialang
   - Metrik Antrian oleh Broker
   - Metrik Topik oleh Broker

   Dalam contoh ini, Metrik Broker dipilih.
5. Kini Anda dapat memeriksa metrik Amazon MQ:

   - Untuk menyortir metrik, gunakan judul kolom.
   - Untuk membuat grafik metrik, pilih kotak centang di samping metrik.
   - Untuk memfilter berdasarkan metrik, pilih nama metrik dan kemudian pilih Tambahkan ke pencarian.

CloudWatch Metrik yang tersedia Amazon MQ untuk broker ActiveMQ

Amazon MQ untuk metrik ActiveMQ

Metrik	Unit	Deskripsi
`AmqpMaximumConnections`	Hitungan	Jumlah maksimum klien yang dapat Anda hubungkan ke broker Anda menggunakan

Metrik	Unit	Deskripsi
		AMQP. Untuk informasi lebih lanjut tentang kuota koneksi, lihatQuotas in Amazon MQ.
`BurstBalance`	Persen	Persentase kredit burst yang tersisa pada volume Amazon EBS digunakan untuk menahan data pesan bagi broker yang dioptimal kan throughput. Jika saldo ini mencapai nol, IOPS yang disediakan oleh volume Amazon EBS akan menurun hingga Saldo Burst diisi ulang. Untuk informasi selengkapnya tentang cara kerja Saldo Burst di Amazon EBS, lihat: Kredit I/O dan Performa Burst.

Metrik	Unit	Deskripsi
`CpuCreditBalance`	Kredit (vCPU-menit)	**Important** Metrik ini hanya tersedia untuk tipe instans broker `mq.t2.micro` . Metrik kredit CPU hanya tersedia pada interval lima menit. Jumlah kredit CPU yang diperoleh dan dikumpulkan instans sejak diluncurkan atau dimulai (termasuk jumlah kredit peluncuran). Saldo kredit tersedia bagi instans broker untuk digunakan di burst di luar pemanfaatan CPU dasar. Kredit dikumpulkan dalam saldo kredit setelah diperoleh , dan dihapus dari saldo kredit setelah digunakan. Saldo kredit memiliki batas maksimum. Setelah batas tercapai, kredit yang baru diperoleh akan dibuang.
`CpuUtilization`	Persen	Persentase unit komputasi Amazon EC2 yang dialokasi kan dan saat ini digunakan oleh broker.

Metrik	Unit	Deskripsi
`CurrentConnectionsCount`	Hitungan	Jumlah koneksi aktif saat ini pada broker saat ini.
`EstablishedConnectionsCount`	Hitungan	Jumlah total koneksi, aktif dan tidak aktif, yang telah ditetapkan pada broker.
`HeapUsage`	Persen	Persentase batas memori ActiveMQ JVM yang saat ini digunakan oleh broker.
`InactiveDurableTopicSubscribersCount`	Hitungan	Jumlah pelanggan topik tahan lama yang tidak aktif, hingga maksimum 2000.
`JobSchedulerStorePercentUsage`	Persen	Persentase ruang disk yang digunakan oleh penyimpanan penjadwal tugas.
`JournalFilesForFastRecovery`	Hitungan	Jumlah file jurnal yang akan diputar ulang setelah penonaktifan bersih.
`JournalFilesForFullRecovery`	Hitungan	Jumlah file jurnal yang akan diputar ulang setelah penonaktifan tidak bersih.
`MqttMaximumConnections`	Hitungan	Jumlah maksimum klien yang dapat Anda hubungkan ke broker Anda menggunakan MQTT. Untuk informasi lebih lanjut tentang kuota koneksi, lihatQuotas in Amazon MQ.

Metrik	Unit	Deskripsi
NetworkConnectorConnectionCount	Hitungan	Jumlah node yang terhubung ke broker dalam jaringan broker menggunakan NetworkConnector.
NetworkIn	Byte	Volume lalu lintas masuk untuk broker.
NetworkOut	Byte	Volume lalu lintas keluar untuk broker.
OpenTransactionCount	Hitungan	Jumlah total transaksi yang sedang berlangsung.
OpenwireMaximumConnections	Hitungan	Jumlah maksimum klien yang dapat Anda hubungkan ke broker Anda menggunakan OpenWire. Untuk informasi lebih lanjut tentang kuota koneksi, lihatQuotas in Amazon MQ.
StompMaximumConnections	Hitungan	Jumlah maksimum klien yang dapat Anda hubungkan ke broker Anda menggunakan STOMP. Untuk informasi lebih lanjut tentang kuota koneksi, lihatQuotas in Amazon MQ.
StorePercentUsage	Persen	Persen yang digunakan oleh batas penyimpanan. Jika ini mencapai 100, broker akan menolak pesan.

Metrik	Unit	Deskripsi
`TempPercentUsage`	Persen	Persentase penyimpanan sementara yang tersedia dan digunakan oleh pesan nonpersisten.
`TotalConsumerCount`	Hitungan	Jumlah konsumen pesan berlangganan ke tujuan pada broker saat ini.
`TotalMessageCount`	Hitungan	Jumlah pesan yang disimpan pada broker.
`TotalProducerCount`	Hitungan	Jumlah produsen pesan aktif di tujuan pada broker saat ini.
`VolumeReadOps`	Hitungan	Jumlah operasi baca yang dilakukan pada volume Amazon EBS.
`VolumeWriteOps`	Hitungan	Jumlah operasi tulis yang dilakukan pada volume Amazon EBS.
`WsMaximumConnections`	Hitungan	Jumlah maksimum klien yang dapat Anda hubungkan ke broker Anda menggunakan WebSocket. Untuk informasi lebih lanjut tentang kuota koneksi, lihatQuotas in Amazon MQ.

Dimensi untuk metrik broker ActiveMQ

Dimensi	Deskripsi
`Broker`	Nama broker

Dimensi	Deskripsi
	Note Broker instans tunggal memiliki sufiks -1. active/standby Broker untuk ketersediaan tinggi memiliki sufiks -1 dan -2 untuk pasangan redundan.

Metrik tujuan ActiveMQ (antrean dan topik)

Important

Metrik berikut mencakup penghitungan per menit untuk periode pemungutan suara. CloudWatch

- `EnqueueCount`
- `ExpiredCount`
- `DequeueCount`
- `DispatchCount`
- `InFlightCount`

Misalnya, dalam CloudWatch periode lima menit, `EnqueueCount` memiliki lima nilai hitungan, masing-masing untuk porsi satu menit dari periode tersebut. Statistik `Minimum` dan `Maximum` memberikan nilai per menit terendah dan tertinggi selama periode tertentu.

Metrik	Unit	Deskripsi
`ConsumerCount`	Hitungan	Jumlah konsumen yang berlangganan ke tujuan.
`EnqueueCount`	Hitungan	Jumlah pesan yang dikirim ke tujuan, per menit.

Metrik	Unit	Deskripsi
`EnqueueTime`	Waktu (milidetik)	end-to-endLatensi dari saat pesan tiba di broker sampai dikirim ke konsumen. **Note** `EnqueueTime` tidak mengukur end-to-end latensi dari saat pesan dikirim oleh produsen sampai mencapai broker, atau latensi dari saat pesan diterima oleh broker sampai diakui oleh broker. Sebaliknya, `EnqueueTime` adalah jumlah milidetik dari saat pesan diterima oleh broker hingga berhasil dikirimkan ke konsumen.
`ExpiredCount`	Hitungan	Jumlah pesan yang tidak dapat dikirimkan karena kedaluwarsa, per menit.
`DispatchCount`	Hitungan	Jumlah pesan yang dikirimkan ke konsumen, per menit.
`DequeueCount`	Hitungan	Jumlah pesan yang diakui oleh konsumen, per menit.

Metrik	Unit	Deskripsi
`InFlightCount`	Hitungan	Jumlah pesan yang dikirimkan ke konsumen dan belum diakui.
`ReceiveCount`	Hitungan	Jumlah pesan yang telah diterima dari broker jauh untuk konektor jaringan dupleks.
`MemoryUsage`	Persen	Persentase batas memori yang saat ini digunakan oleh tujuan.
`ProducerCount`	Hitungan	Jumlah produsen untuk tujuan.
`QueueSize`	Hitungan	Jumlah pesan dalam antrean. **Important** Metrik ini hanya berlaku untuk antrean.
`TotalEnqueueCount`	Hitungan	Jumlah total pesan yang telah dikirimkan ke broker.
`TotalDequeueCount`	Hitungan	Jumlah total pesan yang telah dikonsumsi oleh klien.

Note

Metrik `TotalEnqueueCount` dan `TotalDequeueCount` mencakup pesan untuk topik penasihat. Untuk informasi selengkapnya tentang pesan topik penasihat, lihat Dokumentasi ActiveMQ.

Dimensi untuk metrik tujuan ActiveMQ (antrean dan topik)

Dimensi	Deskripsi
`Broker`	Nama broker. **Note** Broker instans tunggal memiliki sufiks `-1`. active/standby Pialang untuk ketersediaan tinggi memiliki sufiks `-1` dan `-2` untuk pasangan redundan.
`Topic` atau `Queue`	Nama topik atau antrean.
`NetworkConnector`	Nama konektor jaringan.

CloudWatch Metrik yang tersedia untuk Amazon MQ untuk broker RabbitMQ

Metrik broker RabbitMQ

Metrik	Unit	Deskripsi
`ExchangeCount`	Hitungan	Jumlah total pertukaran yang dikonfigurasi pada broker.
`QueueCount`	Hitungan	Jumlah total antrean yang dikonfigurasi pada broker.
`ConnectionCount`	Hitungan	Jumlah total koneksi yang ditetapkan pada broker.
`ChannelCount`	Hitungan	Jumlah total saluran yang ditetapkan pada broker.

Metrik	Unit	Deskripsi
`ConsumerCount`	Hitungan	Jumlah total konsumen yang terhubung ke broker.
`MessageCount`	Hitungan	Jumlah total pesan dalam antrean. **Note** Jumlah yang dihasilkan adalah jumlah total pesan siap dan tidak diakui pada broker.
`MessageReadyCount`	Hitungan	Jumlah total pesan yang siap dalam antrean.
`MessageUnacknowledgedCount`	Hitungan	Jumlah total pesan yang tidak diakui dalam antrean.
`PublishRate`	Hitungan	Laju saat pesan diterbitkan untuk broker. Jumlah yang dihasilkan mewakili jumlah pesan per detik pada saat pengambilan sampel.

Metrik	Unit	Deskripsi
`ConfirmRate`	Hitungan	Laju saat server RabbitMQ mengonfirmasi pesan yang diterbitkan. Anda dapat membandingkan metrik ini dengan `PublishRate` untuk lebih kinerja broker Anda. Jumlah yang dihasilkan mewakili jumlah pesan per detik pada saat pengambilan sampel.
`AckRate`	Hitungan	Laju saat pesan diakui oleh konsumen. Jumlah yang dihasilkan mewakili jumlah pesan per detik pada saat pengambilan sampel.
`SystemCpuUtilization`	Persen	Persentase unit komputasi Amazon EC2 yang dialokasi kan dan saat ini digunakan oleh broker. Untuk penerapan cluster, nilai ini mewakili agregat dari ketiga nilai metrik yang sesuai dari ketiga node RabbitMQ.
`RabbitMQMemLimit`	Byte	Batas RAM untuk broker RabbitMQ. Untuk penerapan cluster, nilai ini mewakili agregat dari ketiga nilai metrik yang sesuai dari ketiga node RabbitMQ.

Metrik	Unit	Deskripsi
`RabbitMQMemUsed`	Byte	Volume RAM yang digunakan oleh broker RabbitMQ. Untuk penerapan cluster, nilai ini mewakili agregat dari ketiga nilai metrik yang sesuai dari ketiga node RabbitMQ.
`RabbitMQDiskFreeLimit`	Byte	Batas disk untuk broker RabbitMQ. Untuk penerapan cluster, nilai ini mewakili agregat dari ketiga nilai metrik yang sesuai dari ketiga node RabbitMQ. Metrik ini berbeda per ukuran instans.
`RabbitMQDiskFree`	Byte	Total volume ruang disk kosong yang tersedia di broker RabbitMQ. Ketika penggunaan disk melampaui batas, klaster akan memblokir semua koneksi produsen. Untuk penerapan cluster, nilai ini mewakili agregat dari ketiga nilai metrik yang sesuai dari ketiga node RabbitMQ.
`RabbitMQFdUsed`	Hitungan	Jumlah deskriptor file yang digunakan. Untuk penerapan cluster, nilai ini mewakili agregat dari ketiga nilai metrik yang sesuai dari ketiga node RabbitMQ.

Metrik	Unit	Deskripsi
RabbitMQIOReadAverageTime	Hitungan	Waktu rata-rata (dalam milidetik) untuk RabbitMQ untuk melakukan satu operasi baca. Nilai sebanding dengan ukuran pesan.
RabbitMQIOWriteAverageTime	Hitungan	Waktu rata-rata (dalam milidetik) untuk RabbitMQ untuk melakukan satu operasi tulis. Nilai sebanding dengan ukuran pesan.

Dimensi untuk metrik broker RabbitMQ

Dimensi	Deskripsi
Broker	Nama broker.

Metrik simpul RabbitMQ

Metrik	Unit	Deskripsi
SystemCpuUtilization	Persen	Persentase unit komputasi Amazon EC2 yang dialokasikan dan saat ini digunakan oleh broker.
RabbitMQMemLimit	Byte	Batas RAM untuk simpul RabbitMQ.
RabbitMQMemUsed	Byte	Volume RAM yang digunakan oleh simpul RabbitMQ. Ketika penggunaan memori melampaui batas, klaster akan

Metrik	Unit	Deskripsi
		memblokir semua koneksi produsen.
`RabbitMQDiskFreeLimit`	Byte	Batas disk untuk simpul RabbitMQ. Metrik ini berbeda per ukuran instans.
`RabbitMQDiskFree`	Byte	Volume total ruang disk kosong yang tersedia di simpul RabbitMQ. Ketika penggunaan disk melampaui batas, klaster akan memblokir semua koneksi produsen.
`RabbitMQFdUsed`	Hitungan	Jumlah deskriptor file yang digunakan.

Dimensi untuk metrik simpul RabbitMQ

Dimensi	Deskripsi
`Node`	Nama simpul. **Note** Nama simpul terdiri dari dua bagian: prefiks (biasanya `rabbit`) dan nama host. Misalnya, `rabbit@ip-10-0-0-230.us-west-2.compute.internal` adalah nama simpul dengan prefiks `rabbit` dan nama host `ip-10-0-0-230.us-west-2.compute.internal`.

Dimensi	Deskripsi
`Broker`	Nama broker.

Metrik antrean RabbitMQ

Metrik	Unit	Deskripsi
`ConsumerCount`	Hitungan	Jumlah konsumen berlangganan ke antrean.
`MessageReadyCount`	Hitungan	Jumlah pesan yang saat ini tersedia untuk dikirimkan.
`MessageUnacknowledgedCount`	Hitungan	Jumlah pesan yang pengakuannya ditunggu oleh server.
`MessageCount`	Hitungan	Jumlah total `MessageReadyCount` dan `MessageUnacknowledgedCount` (juga dikenal sebagai kedalaman antrean).

Dimensi untuk metrik antrean RabbitMQ

> **Note**
>
> Amazon MQ untuk RabbitMQ tidak akan mempublikasikan metrik untuk host virtual dan antrian dengan nama yang berisi spasi kosong, tab, atau karakter non-ASCII lainnya. Untuk informasi selengkapnya tentang nama dimensi, lihat Dimensi di Referensi Amazon CloudWatch API.

Dimensi	Deskripsi
`Queue`	Nama antrean.
`VirtualHost`	Nama host virtual.
`Broker`	Nama broker.

Metrik jaringan RabbitMQ

Metrik	Unit	Deskripsi
`NetworkOut`	Byte	Jumlah bita yang dikirimkan oleh instans di semua antarmuka jaringan. Metrik ini mengidentifikasi volume lalu lintas jaringan yang keluar dari instans tunggal. Jumlah yang dilaporkan adalah jumlah bita yang dikirimkan selama periode tersebut. Jika menggunakan pemantauan dasar (5 menit) dan statistiknya adalah Sum, Anda dapat membagi jumlah ini dengan 300 untuk menemukan Bita/detik. Jika Anda memiliki pemantauan terperinci (1 menit) dan statistik nya adalah Sum, bagilah dengan 60. Anda juga dapat menggunakan fungsi matematika CloudWatch metrik `DIFF_TIME` untuk menemukan byte per detik. Misalnya, jika Anda telah membuat grafik NetworkOut CloudWatc h sebagai`m1`, rumus matematika metrik `m1/(DIFF_ TIME(m1))` mengembalikan metrik dalam byte/deti k. Untuk informasi selengkapnya tentang `DIFF_TIME` dan fungsi matematika metrik lainnya, lihat Menggunakan matematika metrik. Statistik Berarti: Jumlah, Rata-rata, Minimum, Maksimum
`NetworkIn`	Byte	Jumlah bita yang diterima oleh instans di semua antarmuka jaringan. Metrik ini mengidentifikasi volume lalu lintas jaringan yang masuk ke instans tunggal. Jumlah yang dilaporkan adalah jumlah bita yang diterima selama periode

Metrik	Unit	Deskripsi
		tersebut. Jika menggunakan pemantauan dasar (5 menit) dan statistiknya adalah Sum, Anda dapat membagi jumlah ini dengan 300 untuk menemukan Bita/detik. Jika Anda memiliki pemantauan terperinci (1 menit) dan statistik nya adalah Sum, bagilah dengan 60. Anda juga dapat menggunakan fungsi matematika CloudWatch metrik `DIFF_TIME` untuk menemukan byte per detik. Misalnya, jika Anda telah membuat grafik NetworkIn CloudWatc h sebagai`m1`, rumus matematika metrik `m1/(DIFF_ TIME(m1))` mengembalikan metrik dalam byte/deti k. Untuk informasi selengkapnya tentang `DIFF_TIME` dan fungsi matematika metrik lainnya, lihat Menggunakan matematika metrik. Statistik Berarti: Jumlah, Rata-rata, Minimum, Maksimum

Dimensi untuk broker RabbitMQ

Dimensi	Deskripsi
`BrokerId`	Id dari broker

Mengkonfigurasi Amazon MQ untuk log RabbitMQ

Saat Anda mengaktifkan CloudWatch pencatatan untuk broker RabbitMQ Anda, Amazon MQ menggunakan peran terkait layanan untuk mempublikasikan log umum. CloudWatch Jika tidak ada peran terkait layanan Amazon MQ saat Anda pertama kali membuat broker, Amazon MQ akan membuatnya secara otomatis. Semua broker RabbitMQ berikutnya akan menggunakan peran terkait layanan yang sama untuk mempublikasikan log ke. CloudWatch

Untuk informasi selengkapnya tentang peran terkait layanan, lihat Menggunakan peran terkait layanan di Panduan Pengguna.AWS Identity and Access Management Untuk informasi selengkapnya tentang cara Amazon MQ menggunakan peran terkait layanan, lihat the section called “Menggunakan peran yang terhubung dengan layanan”.

Mencatat panggilan Amazon MQ API menggunakan AWS CloudTrail

Amazon MQ terintegrasi dengan AWS CloudTrail, layanan yang menyediakan catatan panggilan Amazon MQ yang dibuat oleh pengguna, peran, AWS atau layanan. CloudTrail menangkap panggilan API yang terkait dengan broker dan konfigurasi Amazon MQ sebagai acara, termasuk panggilan dari konsol Amazon MQ dan panggilan kode dari Amazon MQ. APIs Untuk informasi selengkapnya CloudTrail, lihat Panduan AWS CloudTrail Pengguna.

> **Note**
>
> CloudTrail tidak mencatat panggilan API yang terkait dengan operasi ActiveMQ (misalnya, mengirim dan menerima pesan) atau ke ActiveMQ Web Console. Untuk mencatat informasi yang terkait dengan operasi ActiveMQ, Anda dapat mengonfigurasi Amazon MQ untuk mempublikasikan log umum dan audit ke Amazon Logs. CloudWatch

Dengan menggunakan informasi yang CloudTrail dikumpulkan, Anda dapat mengidentifikasi permintaan khusus ke Amazon MQ API, alamat IP pemohon, identitas pemohon, tanggal dan waktu permintaan, dan sebagainya. Jika mengonfigurasi jejak, Anda dapat mengaktifkan pengiriman CloudTrail acara secara terus menerus ke bucket Amazon S3. Jika Anda tidak mengonfigurasi jejak, Anda dapat melihat peristiwa terbaru dalam riwayat acara di CloudTrail konsol. Untuk informasi selengkapnya, lihat Gambaran Umum Pembuatan Jejak di Panduan Pengguna AWS CloudTrail.

Informasi Amazon MQ di CloudTrail

Saat Anda membuat AWS akun, CloudTrail diaktifkan. Saat aktivitas acara Amazon MQ yang didukung terjadi, aktivitas tersebut direkam dalam CloudTrail peristiwa dengan peristiwa AWS layanan lain dalam riwayat acara. Anda dapat melihat, mencari, dan mengunduh kejadian terbaru untuk akun AWS Anda. Untuk informasi selengkapnya, lihat Melihat CloudTrail Acara dengan Riwayat Acara di Panduan AWS CloudTrail Pengguna.

Jejak memungkinkan CloudTrail untuk mengirimkan file log ke bucket Amazon S3. Anda dapat membuat jejak untuk menyimpan catatan peristiwa yang sedang berlangsung di AWS akun Anda. Secara default, saat Anda membuat jejak menggunakan Konsol Manajemen AWS, jejak tersebut berlaku untuk semua AWS Wilayah. Jejak mencatat peristiwa dari semua AWS Wilayah dan mengirimkan file log ke bucket Amazon S3 yang ditentukan. Anda juga dapat mengonfigurasi AWS layanan lain untuk menganalisis lebih lanjut dan bertindak atas data peristiwa yang dikumpulkan

dalam CloudTrail log. Untuk informasi selengkapnya, lihat topik berikut di Panduan Pengguna AWS CloudTrail :

- CloudTrail Layanan dan Integrasi yang Didukung
- Mengkonfigurasi Notifikasi Amazon SNS untuk CloudTrail
- Menerima File CloudTrail Log dari Beberapa Wilayah
- Menerima File CloudTrail Log dari Beberapa Akun

Amazon MQ mendukung pencatatan parameter permintaan dan tanggapan untuk hal berikut APIs sebagai peristiwa dalam file CloudTrail log:

- `CreateConfiguration`
- `DeleteBroker`
- `DeleteUser`
- `RebootBroker`
- `UpdateBroker`

> **Note**
>
> RebootBroker file log dicatat saat Anda me-reboot broker. Selama jendela pemeliharaan, layanan secara otomatis reboot, dan file RebootBroker log tidak dicatat.

> **Important**
>
> Untuk GET metode berikut ini APIs, parameter permintaan dicatat, tetapi tanggapannya disunting:
>
> - `DescribeBroker`
> - `DescribeConfiguration`
> - `DescribeConfigurationRevision`
> - `DescribeUser`
> - `ListBrokers`
> - `ListConfigurationRevisions`

- `ListConfigurations`
- `ListUsers`

Untuk hal berikut APIs, parameter `password` permintaan `data` dan disembunyikan oleh tanda bintang ()`***`:

- `CreateBroker` (POST)
- `CreateUser` (POST)
- `UpdateConfiguration` (PUT)
- `UpdateUser` (PUT)

Setiap entri kejadian atau log berisi informasi tentang peminta. Informasi ini membantu Anda menentukan hal berikut:

- Apakah permintaan dibuat dengan kredensi root atau pengguna?
- Apakah permintaan dibuat dengan kredensial keamanan sementara untuk peran atau pengguna gabungan?
- Apakah permintaan itu dibuat oleh AWS layanan lain?

Untuk informasi selengkapnya, lihat Elemen CloudTrail UserIdentity di AWS CloudTrail Panduan Pengguna.

Contoh Entri Berkas Log Amazon MQ

Trail adalah konfigurasi yang memungkinkan pengiriman peristiwa sebagai file log ke bucket Amazon S3 yang ditentukan. CloudTrail file log berisi satu atau lebih entri log.

Kejadian mewakili permintaan tunggal dari sumber apa pun dan mencakup informasi tentang permintaan ke API Amazon MQ, alamat IP peminta, identitas peminta, tanggal serta waktu permintaan, dan sebagainya.

Contoh berikut menunjukkan entri CloudTrail log untuk panggilan `CreateBroker`API.

> **Note**
>
> Karena file CloudTrail log bukan merupakan jejak tumpukan publik yang diurutkan APIs, mereka tidak mencantumkan informasi dalam urutan tertentu.


```
{
    "eventVersion": "1.06",
    "userIdentity": {
        "type": "IAMUser",
        "principalId": "AKIAIOSFODNN7EXAMPLE",
        "arn": "arn:aws:iam::111122223333:user/AmazonMqConsole",
        "accountId": "111122223333",
        "accessKeyId": "AKIAI44QH8DHBEXAMPLE",
        "userName": "AmazonMqConsole"
    },
    "eventTime": "2018-06-28T22:23:46Z",
    "eventSource": "amazonmq.amazonaws.com",
    "eventName": "CreateBroker",
    "awsRegion": "us-west-2",
    "sourceIPAddress": "203.0.113.0",
    "userAgent": "PostmanRuntime/7.1.5",
    "requestParameters": {
        "engineVersion": "5.15.9",
        "deploymentMode": "ACTIVE_STANDBY_MULTI_AZ",
        "maintenanceWindowStartTime": {
            "dayOfWeek": "THURSDAY",
            "timeOfDay": "22:45",
            "timeZone": "America/Los_Angeles"
        },
        "engineType": "ActiveMQ",
        "hostInstanceType": "mq.m5.large",
        "users": [
            {
                "username": "MyUsername123",
                "password": "***",
                "consoleAccess": true,
                "groups": [
                    "admins",
                    "support"
                ]
            },
```

```
            {
                "username": "MyUsername456",
                "password": "***",
                "groups": [
                    "admins"
                ]
            }
        ],
        "creatorRequestId": "1",
        "publiclyAccessible": true,
        "securityGroups": [
            "sg-a1b234cd"
        ],
        "brokerName": "MyBroker",
        "autoMinorVersionUpgrade": false,
        "subnetIds": [
            "subnet-12a3b45c",
            "subnet-67d8e90f"
        ]
    },
    "responseElements": {
        "brokerId": "b-1234a5b6-78cd-901e-2fgh-3i45j6k178l9",
        "brokerArn": "arn:aws:mq:us-
east-2:123456789012:broker:MyBroker:b-1234a5b6-78cd-901e-2fgh-3i45j6k178l9"
    },
    "requestID": "a1b2c345-6d78-90e1-f2g3-4hi56jk7l890",
    "eventID": "a12bcd3e-fg45-67h8-ij90-12k34d5l16mn",
    "readOnly": false,
    "eventType": "AwsApiCall",
    "recipientAccountId": "111122223333"
}
```


Mengkonfigurasi Amazon MQ untuk log ActiveMQ

Untuk mengizinkan Amazon MQ mempublikasikan CloudWatch log ke Log, Anda harus menambahkan izin ke pengguna Amazon MQ Anda dan juga mengonfigurasi kebijakan berbasis sumber daya untuk Amazon MQ sebelum Anda membuat atau memulai ulang broker.

> **Note**
>
> Saat Anda mengaktifkan log dan mempublikasikan pesan dari konsol web ActiveMQ, konten pesan dikirim CloudWatch ke dan ditampilkan di log.

Berikut ini menjelaskan langkah-langkah untuk mengkonfigurasi CloudWatch log untuk broker ActiveMQ Anda.

Topik

- Memahami struktur logging di CloudWatch Log
- Tambahkan CreateLogGroup izin ke pengguna Amazon MQ Anda
- Mengonfigurasi kebijakan berbasis sumber daya untuk Amazon MQ
- Pencegahan "confused deputy" lintas layanan

Memahami struktur logging di CloudWatch Log

Anda dapat mengaktifkan pencatatan umum dan audit saat Anda mengonfigurasi pengaturan broker tingkat lanjut saat Anda membuat broker, atau saat Anda mengedit broker.

Pencatatan umum memungkinkan tingkat `INFO` pencatatan default (`DEBUG`pencatatan tidak didukung) dan `activemq.log` memublikasikan ke grup log di CloudWatch akun Anda. Grup log memiliki format yang serupa dengan hal berikut:

```
/aws/amazonmq/broker/b-1234a5b6-78cd-901e-2fgh-3i45j6k178l9/general
```

Pencatatan audit memungkinkan pencatatan tindakan manajemen yang dilakukan menggunakan JMX atau menggunakan ActiveMQ Web Console dan memublikasikan `audit.log` ke grup log di akun Anda. CloudWatch Grup log memiliki format yang serupa dengan hal berikut:

```
/aws/amazonmq/broker/b-1234a5b6-78cd-901e-2fgh-3i45j6k178l9/audit
```

Tergantung pada apakah Anda memiliki broker instans tunggal atau broker aktif/siaga, Amazon MQ membuat satu atau dua pengaliran log dalam setiap grup log. Pengaliran log memiliki format yang serupa dengan hal berikut.

```
activemq-b-1234a5b6-78cd-901e-2fgh-3i45j6k178l9-1.log
```

```
activemq-b-1234a5b6-78cd-901e-2fgh-3i45j6k178l9-2.log
```

Sufiks `-1` dan `-2` menunjukkan instans broker individual. Untuk informasi selengkapnya, lihat Bekerja dengan Grup Log dan Aliran Log di Panduan Pengguna Amazon CloudWatch Logs.

Tambahkan **`CreateLogGroup`** izin ke pengguna Amazon MQ Anda

Untuk mengizinkan Amazon MQ membuat grup CloudWatch log Log, Anda harus memastikan bahwa pengguna yang membuat atau me-reboot broker memiliki izin. `logs:CreateLogGroup`

> **Important**
>
> Jika Anda tidak menambahkan izin `CreateLogGroup` ke pengguna Amazon MQ sebelum pengguna membuat atau mem-boot ulang broker, Amazon MQ tidak membuat grup log.

Contoh kebijakan berbasis IAM berikut memberikan izin untuk `logs:CreateLogGroup` bagi pengguna yang dilampirkan kebijakan ini.

JSON

```
{
                    "Version":"2012-10-17",
                    "Statement": [
                        {
                            "Effect": "Allow",
                            "Action": "logs:CreateLogGroup",
                            "Resource": "arn:aws:logs:*:*:log-group:/aws/
amazonmq/*"
                        }
                    ]
                    }
```

> **Note**
>
> Di sini, istilah pengguna mengacu pada Pengguna dan bukan pengguna Amazon MQ, yang dibuat ketika broker baru dikonfigurasi. Untuk informasi lebih lanjut mengenai pengaturan

pengguna dan konfigurasi kebijakan IAM, silakan merujuk ke bagian Ikhtisar Manajemen Identitas dari Panduan Pengguna IAM.

Untuk informasi selengkapnya, lihat `CreateLogGroup` di Referensi API Amazon CloudWatch Logs.

Mengonfigurasi kebijakan berbasis sumber daya untuk Amazon MQ

Important

Jika Anda tidak mengonfigurasi kebijakan berbasis sumber daya untuk Amazon MQ, broker tidak dapat mempublikasikan log ke Log. CloudWatch

Untuk mengizinkan Amazon MQ memublikasikan log ke grup CloudWatch log Log Anda, konfigurasikan kebijakan berbasis sumber daya untuk memberi Amazon MQ akses ke tindakan API Log berikut: CloudWatch

- `CreateLogStream`— Membuat aliran CloudWatch log Log untuk grup log tertentu.
- `PutLogEvents`— Mengirimkan peristiwa ke aliran CloudWatch log Log yang ditentukan.

Kebijakan berbasis sumber daya berikut memberikan izin untuk dan untuk. `logs:CreateLogStream` `logs:PutLogEvents` AWS

JSON

```
{
                        "Version":"2012-10-17",
                        "Statement": [
                            {
                                "Effect": "Allow",
                                "Principal": { "Service":
 "mq.amazonaws.com" },
                                "Action": [ "logs:CreateLogStream",
 "logs:PutLogEvents" ],
                                "Resource": "arn:aws:logs:*:*:log-group:/aws/
amazonmq/*"
                            }
```

```
                ]
            }
```

Kebijakan berbasis sumber daya ini harus dikonfigurasi dengan menggunakan AWS CLI seperti yang ditunjukkan oleh perintah berikut. Dalam contoh, ganti *us-east-1* dengan informasi Anda sendiri.

```
aws --region us-east-1 logs put-resource-policy --policy-name AmazonMQ-logs \
                        --policy-document "{\"Version\": \"2012-10-17\", \"Statement\":
[{ \"Effect\": \"Allow\", \"Principal\": { \"Service\": \"mq.amazonaws.com\" },
                        \"Action\": [\"logs:CreateLogStream\", \"logs:PutLogEvents\"],
 \"Resource\": \"arn:aws:logs:*:*:log-group:\/aws\/amazonmq\/*\" }]}"
```

> **Note**
>
> Karena contoh ini menggunakan `/aws/amazonmq/` awalan, Anda perlu mengonfigurasi kebijakan berbasis sumber daya hanya sekali per akun, per AWS wilayah.

Pencegahan "confused deputy" lintas layanan

Masalah "confused deputy" adalah masalah keamanan di mana entitas yang tidak memiliki izin untuk melakukan tindakan dapat memengaruhi entitas yang memiliki hak akses lebih tinggi untuk melakukan tindakan. Pada tahun AWS, peniruan lintas layanan dapat mengakibatkan masalah wakil yang membingungkan. Peniruan identitas lintas layanan dapat terjadi ketika satu layanan (layanan yang dipanggil) memanggil layanan lain (layanan yang dipanggil). Layanan pemanggilan dapat dimanipulasi menggunakan izinnya untuk bertindak pada sumber daya pelanggan lain dengan cara yang seharusnya tidak dilakukannya kecuali bila memiliki izin untuk mengakses. Untuk mencegah hal ini, AWS sediakan alat yang membantu Anda melindungi data Anda untuk semua layanan dengan prinsip layanan yang telah diberikan akses ke sumber daya di akun Anda.

Sebaiknya gunakan kunci konteks kondisi `aws:SourceAccount` global `aws:SourceArn` dan global dalam kebijakan berbasis sumber daya Amazon MQ Anda untuk membatasi akses CloudWatch Log ke satu atau beberapa broker tertentu.

> **Note**
>
> Jika Anda menggunakan kedua kunci konteks kondisi global, `aws:SourceAccount` nilai dan akun dalam `aws:SourceArn` nilai harus menggunakan ID akun yang sama saat digunakan dalam pernyataan kebijakan yang sama.

Contoh berikut menunjukkan kebijakan berbasis sumber daya yang membatasi akses CloudWatch Log ke satu broker MQ Amazon.

JSON

```
{
                        "Version":"2012-10-17",
                        "Statement": [
                            {
                            "Effect": "Allow",
                            "Principal": {
                                "Service": "mq.amazonaws.com"
                            },
                            "Action": [
                                "logs:CreateLogStream",
                                "logs:PutLogEvents"
                            ],
                            "Resource": "arn:aws:logs:*:*:log-group:/aws/
amazonmq/*",
                            "Condition": {
                                "StringEquals": {
                                "aws:SourceAccount": "123456789012",
                                "aws:SourceArn": "arn:aws:mq:us-
west-1:123456789012:broker:my-broker:123456789012"
                                }
                            }
                            }
                        ]
                        }
```

Anda juga dapat mengonfigurasi kebijakan berbasis sumber daya Anda untuk membatasi akses CloudWatch Log ke semua broker di akun, seperti yang ditunjukkan di bawah ini.

JSON

```
{
                            "Version":"2012-10-17",
                            "Statement": [
                            {
                                "Effect": "Allow",
                                "Principal": {
                                "Service": [
                                    "mq.amazonaws.com"
                                ]
                                },
                                "Action": [
                                "logs:CreateLogStream",
                                "logs:PutLogEvents"
                                ],
                                "Resource": "arn:aws:logs:*:*:log-group:/aws/
amazonmq/*",
                                "Condition": {
                                "ArnLike": {
                                    "aws:SourceArn":
 "arn:aws:mq:*:123456789012:broker:*"
                                },
                                "StringEquals": {
                                    "aws:SourceAccount": "123456789012"
                                }
                                }
                            }
                            ]
                        }
```

Untuk informasi lebih lanjut tentang masalah keamanan wakil yang membingungkan, lihat Masalah wakil yang bingung di Panduan Pengguna.

Mengatasi Masalah Konfigurasi CloudWatch Log dengan Amazon MQ

Dalam beberapa kasus, CloudWatch Log mungkin tidak selalu berperilaku seperti yang diharapkan. Bagian ini memberikan gambaran umum dari masalah umum dan menunjukkan cara mengatasinya.

Grup Log Tidak Muncul di CloudWatch

Tambahkan izin `CreateLogGroup` ke pengguna Amazon MQ Anda dan boot ulang broker. Hal ini memungkinkan Amazon MQ untuk membuat grup log.

Aliran Log Tidak Muncul di Grup CloudWatch Log

Konfigurasi kebijakan berbasis sumber daya untuk Amazon MQ. Hal ini memungkinkan broker Anda memublikasikan log-nya.

Kuota di Amazon MQ

Topik ini mencantumkan batas dalam Amazon MQ. Banyak dari batasan berikut dapat diubah untuk AWS akun tertentu. Untuk meminta peningkatan batas, lihat AWS Service Quotas di. Referensi Umum Amazon Web Batas yang diperbarui tidak akan terlihat bahkan setelah kenaikan batas telah diterapkan. Untuk informasi selengkapnya tentang melihat batas koneksi saat ini di Amazon CloudWatch, lihat Memantau broker Amazon MQ menggunakan Amazon. CloudWatch

Topik

- Pialang
- Konfigurasi
- Pengguna
- Penyimpanan Data
- Throttling API

Pialang

Tabel berikut mencantumkan kuota yang terkait dengan broker Amazon MQ.

Kuota	Deskripsi
Nama broker	• Harus unik di AWS akun Anda. • Harus terdiri dari 1-50 karakter. • Hanya dapat berisi karakter yang ditentukan dalam Set Karakter ASCII yang Dapat Dicetak. • Hanya dapat berisi karakter alfanumerik, tanda hubung, titik, garis bawah, dan tilde (- . _ ~).
Jumlah broker, per wilayah	50

Kuota	Deskripsi
Koneksi tingkat kabel per protokol untuk broker yang lebih kecil	**Important** Tidak berlaku untuk broker RabbitMQ. 300 `mq.*.micro` misalnya jenis broker.
Koneksi tingkat kabel per protokol untuk broker yang lebih besar	**Important** Tidak berlaku untuk broker RabbitMQ. 2.000 `mq.*.*large` misalnya jenis broker.
Grup keamanan per broker	5
Tujuan ActiveMQ (antrian, dan topik) dipantau CloudWatch	CloudWatch hanya memonitor 1000 tujuan pertama.
Tujuan RabbitMQ (antrian) dipantau di CloudWatch	CloudWatch memantau hanya 500 tujuan pertama, dipesan berdasarkan jumlah konsumen.
Tanda per broker	50

Konfigurasi

Tabel berikut mencantumkan kuota yang terkait dengan konfigurasi Amazon MQ.

Kuota	Deskripsi
Nama konfigurasi	• Harus terdiri dari 1-150 karakter. •

Kuota	Deskripsi
	Hanya dapat berisi karakter yang ditentukan dalam Set Karakter ASCII yang Dapat Dicetak. • Hanya dapat berisi karakter alfanumerik, tanda hubung, titik, garis bawah, dan tilde (- . _ ~).
Revisi per konfigurasi	300

Pengguna

Tabel berikut mencantumkan kuota yang terkait dengan pengguna broker ActiveMQ Amazon MQ.

Kuota	Deskripsi
Nama pengguna	• Harus terdiri dari 1-100 karakter. • Hanya dapat berisi karakter yang ditentukan dalam Set Karakter ASCII yang Dapat Dicetak. • Hanya dapat berisi karakter alfanumerik, tanda hubung, titik, garis bawah, dan tilde (- . _ ~). • Tidak boleh berisi tanda koma (,).
Kata Sandi	• Harus terdiri dari 12-250 karakter. • Hanya dapat berisi karakter yang ditentukan dalam Set Karakter ASCII yang Dapat Dicetak.

Kuota	Deskripsi
	• Harus terdiri dari setidaknya 4 karakter unik. • Tidak boleh berisi tanda koma (`,`).
Pengguna per broker (auth sederhana)	250
Grup per pengguna (auth sederhana)	20

Penyimpanan Data

Tabel berikut mencantumkan kuota yang terkait dengan penyimpanan data Amazon MQ.

Kuota	Deskripsi
Kapasitas penyimpanan per broker yang lebih kecil	20 GB untuk broker tipe instans `mq.*.micro` . Untuk informasi selengkapnya mengenai tipe instans Amazon MQ, lihat Broker instance types.
Kapasitas penyimpanan per broker yang lebih besar	200 GB untuk broker tipe instans `mq.m5.*`. Untuk informasi selengkapnya mengenai tipe instans Amazon MQ, lihat Broker instance types.
Batasan penggunaan penjadwal tugas per broker didukung Amazon EBS	**Important** Tidak berlaku untuk broker RabbitMQ. 50 GB. Untuk informasi selengkapnya tentang penggunaan penjadwal tugas, lihat `JobSchedulerUsage` dalam Dokumentasi API Apache ActiveMQ.

Kuota	Deskripsi
Kapasitas penyimpanan sementara per broker yang lebih kecil.	**Important** Tidak berlaku untuk broker RabbitMQ. 5 GB untuk broker tipe instans `mq.*.micro` .
Kapasitas penyimpanan sementara per broker yang lebih besar.	**Important** Tidak berlaku untuk broker RabbitMQ. 50 GB untuk broker tipe instans `mq.m5.*`.

Throttling API

Kuota pembatasan berikut dikumpulkan per AWS akun, di semua Amazon MQ untuk mempertahankan bandwidth layanan. APIs Untuk informasi selengkapnya tentang Amazon MQ APIs, lihat Referensi API Amazon MQ REST.

Important

Kuota ini tidak berlaku untuk Amazon MQ untuk ActiveMQ atau Amazon MQ untuk perpesanan broker RabbitMQ. APIs Misalnya, Amazon MQ tidak men-throttle pengiriman atau penerimaan pesan.

Batasan burst API	Batasan laju API
100	15

Memecahkan Masalah Amazon MQ

Bagian ini menjelaskan masalah umum yang mungkin Anda temui saat menggunakan broker Amazon MQ, dan langkah-langkah yang dapat Anda ambil untuk mengatasinya. Untuk pemecahan masalah umum, lihat. the section called “Pemecahan Masalah: Amazon MQ Umum” Untuk memecahkan masalah versi mesin spesifik Anda, lihat bagian berikut.

Memecahkan masalah ActiveMQ di Amazon MQ

Topik pemecahan masalah	Deskripsi
Pemecahan masalah umum	Gunakan informasi di bagian ini untuk membantu Anda mendiagnosis dan menyelesaikan masalah umum yang mungkin Anda temui saat bekerja dengan ActiveMQ di broker Amazon MQ.
BROKER_ENI_DIHAPUS	ActiveMQ di Amazon MQ akan memunculkan alarm saat Anda menghapus `BROKER_ENI_DELETED` Antarmuka Jaringan Elastis (ENI) broker.
BROKER_OOM	ActiveMQ di Amazon MQ akan menaikkan alarm BROKER_OOM ketika broker mengalami loop restart karena kapasitas memori yang tidak mencukupi

Memecahkan masalah RabbitMQ di Amazon MQ

Topik pemecahan masalah	Deskripsi
Pemecahan masalah umum	Diagnosis masalah umum yang mungkin Anda temui saat bekerja dengan broker RabbitMQ.

Topik pemecahan masalah	Deskripsi
RABBITMQ_MEMORY_ALARM	RabbitMQ akan meningkatkan alarm memori tinggi ketika penggunaan memori broker, diidentifikasi dengan CloudWatch metrik`RabbitMQMemUsed` , melebihi batas memori, diidentifikasi oleh. `RabbitMQMemLimit`
RABBITMQ_INVALID_KMS_KEY	RabbitMQ di Amazon MQ akan memunculkan kode yang diperlukan tindakan kritis INVALID_KMS_KEY saat broker yang dibuat dengan pelanggan yang dikelola AWS KMS key(CMK) mendeteksi bahwa kunci (KMS) dinonaktifkan. AWS Key Management Service
RABBITMQ_INVALID_ASSUMEROLE	RabbitMQ di Amazon MQ akan memunculkan kode yang diperlukan tindakan kritis INVALID_ASSUMEROLE saat peran IAM yang ditentukan ARN tidak dapat diasumsikan oleh Amazon MQ. `aws.arns.assume_role_arn`

Topik pemecahan masalah	Deskripsi
RABBITMQ_INVALID_ARN_LDAP	RabbitMQ di Amazon MQ akan memunculkan kode tindakan kritis INVALID_ARN_LDAP yang diperlukan ketika kata sandi akun layanan LDAP ARN tidak valid atau tidak dapat diakses.
RABBITMQ_INVALID_ARN_HTTP	RabbitMQ di Amazon MQ akan memunculkan kode tindakan kritis INVALID_ARN_HTTP yang diperlukan ketika satu atau ARNs beberapa sertifikat SSL atau file kunci untuk HTTP auth_backend tidak valid atau tidak dapat diakses.
RABBITMQ_INVALID_ARN_SSL	RabbitMQ di Amazon MQ akan memunculkan kode yang diperlukan tindakan kritis INVALID_ARN_SSL ketika satu atau beberapa truststore sertifikat CA untuk auth_mechanism EKSTERNAL tidak valid atau tidak dapat diakses. ARNs
RABBITMQ_INVALID_ARN	RabbitMQ di Amazon MQ akan memunculkan kode tindakan kritis INVALID_ARN yang diperlukan ketika satu atau lebih dalam konfigurasi broker tidak valid atau tidak dapat diakses. ARNs

Topik pemecahan masalah	Deskripsi
RABBITMQ_DISK_ALARM	Alarm batas disk adalah indikasi bahwa volume disk yang digunakan oleh node RabbitMQ telah menurun karena tingginya jumlah pesan yang tidak dikonsumsi saat pesan baru ditambahkan.

Pemecahan Masalah: Amazon MQ Umum

Gunakan informasi di bagian ini untuk membantu Anda mendiagnosis masalah umum yang mungkin Anda temui saat bekerja dengan broker Amazon MQ, seperti masalah yang menghubungkan ke broker Anda, dan reboot broker.

Daftar Isi

- Saya tidak dapat terhubung ke konsol web broker atau titik akhir saya.
- Broker saya sedang berjalan, dan saya dapat memverifikasi konektivitas menggunakantelnet, tetapi klien saya tidak dapat terhubung dan mengembalikan pengecualian SSL.
- Saya membuat broker tetapi pembuatan broker gagal.
- Broker saya memulai kembali dan saya tidak yakin mengapa.

Saya tidak dapat terhubung ke konsol web broker atau titik akhir saya.

Jika Anda mengalami masalah saat terhubung ke broker Anda menggunakan konsol web atau titik akhir tingkat kabel, kami merekomendasikan langkah-langkah berikut.

1. Periksa apakah Anda mencoba terhubung ke broker Anda dari balik firewall. Anda mungkin perlu mengkonfigurasi firewall untuk memungkinkan akses ke broker Anda.
2. Periksa apakah Anda mencoba terhubung ke broker Anda menggunakan titik akhir FIPS. Amazon MQ hanya mendukung titik akhir FIPS saat menggunakan operasi API, dan bukan untuk koneksi tingkat kabel ke instance broker itu sendiri.
3. Periksa apakah opsi Aksesibilitas Publik untuk broker Anda diatur ke Ya. Jika ini diatur ke Tidak, periksa aturan Daftar Kontrol Akses (ACL) jaringan subnet Anda. Jika Anda telah membuat

jaringan khusus ACLs, Anda mungkin perlu mengubah aturan ACL jaringan untuk menyediakan akses ke broker Anda. Untuk informasi selengkapnya tentang jaringan VPC Amazon, lihat Mengaktifkan akses internet di Panduan Pengguna Amazon VPC

4. Periksa aturan Grup Keamanan broker Anda. Pastikan Anda mengizinkan koneksi ke port berikut:

> **Note**
>
> Port berikut dikelompokkan menurut jenis mesin karena ActiveMQ di Amazon MQ dan RabbitMQ di Amazon MQ menggunakan port yang berbeda untuk koneksi.

ActiveMQ di Amazon MQ

- Konsol web — Pelabuhan 8162
- OpenWire — Pelabuhan 61617
- AMQP - Pelabuhan 5671
- STOMP - Pelabuhan 61614
- MQTT - Pelabuhan 8883
- WSS - Pelabuhan 61619

RabbitMQ di Amazon MQ

- Konsol web dan API manajemen — Port 443 dan 15671
- AMQP - Pelabuhan 5671

5. Jalankan tes konektivitas jaringan berikut untuk jenis mesin broker Anda.

> **Note**
>
> Untuk broker tanpa aksesibilitas publik, jalankan pengujian dari instans Amazon EC2 dalam VPC Amazon yang sama dengan broker Amazon MQ Anda dan evaluasi tanggapannya.

ActiveMQ on Amazon MQ

Untuk menguji ActiveMQ Anda di konektivitas jaringan broker Amazon MQ

1. Buka terminal baru atau jendela baris perintah.

2. Jalankan `nslookup` perintah berikut untuk menanyakan catatan DNS broker Anda. Untuk penerapan aktif/siaga, uji titik akhir aktif dan siaga. active/standby Titik akhir diidentifikasi dengan akhiran, `-1` atau `-2` ditambahkan ke ID broker unik. Ganti titik akhir dengan informasi Anda.

```
$ nslookup b-1234a5b6-78cd-901e-2fgh-3i45j6k178l9-1.mq.us-west-2.amazonaws.com
```

Jika kueri berhasil, Anda akan melihat output yang mirip dengan berikut ini.

```
Non-authoritative answer:
Server:  dns-resolver-corp-sfo-1.sfo.corp.amazon.com
Address:  172.10.123.456

Name:    ec2-12-345-123-45.us-west-2.compute.amazonaws.com
Address:  12.345.123.45
Aliases:  b-1234a5b6-78cd-901e-2fgh-3i45j6k178l9-1.mq.us-west-2.amazonaws.com
```

Alamat IP yang diselesaikan harus sesuai dengan alamat IP yang disediakan di konsol Amazon MQ. Ini menunjukkan bahwa nama domain diselesaikan dengan benar di server DNS, dan Anda dapat melanjutkan ke langkah berikutnya.

3. Jalankan `telnet` perintah berikut untuk menguji jalur jaringan untuk broker Anda. Ganti titik akhir dengan informasi Anda. Ganti *`port`* dengan nomor port 8162 untuk konsol web, atau port tingkat kabel lainnya untuk menguji protokol tambahan sesuai kebutuhan.

> **Note**
>
> Untuk active/standby penerapan, Anda akan menerima pesan `Connect failed` kesalahan jika Anda menjalankan `telnet` dengan titik akhir siaga. Ini diharapkan, karena instance siaga itu sendiri sedang berjalan, tetapi proses ActiveMQ tidak berjalan dan tidak memiliki akses ke volume penyimpanan Amazon EFS broker. Jalankan perintah untuk keduanya `-1` dan `-2` titik akhir untuk memastikan Anda menguji instance aktif dan siaga.

```
$ telnet b-1234a5b6-78cd-901e-2fgh-3i45j6k178l9-1.mq.us-
west-2.amazonaws.com port
```

Untuk instance aktif, Anda akan melihat output yang mirip dengan berikut ini.

```
Connected to b-1234a5b6-78cd-901e-2fgh-3i45j6k178l9-1.mq.us-
west-2.amazonaws.com.
Escape character is '^]'.
```

4. Lakukan salah satu dari berikut ini.

   - Jika `telnet` perintah berhasil, periksa `EstablishedConnectionsCount`metrik dan konfirmasikan bahwa broker belum mencapai batas koneksi tingkat kabel maksimum. Anda juga dapat mengonfirmasi apakah batas telah tercapai dengan meninjau `General` log broker. Jika metrik ini lebih besar dari nol, maka setidaknya ada satu klien yang saat ini terhubung ke broker. Jika metrik menunjukkan koneksi nol, maka lakukan tes `telnet` jalur lagi dan tunggu setidaknya satu menit sebelum memutuskan sambungan, karena metrik broker diterbitkan setiap menit.
   - Jika `telnet` perintah gagal, periksa status elastic network interface broker Anda, dan konfirmasikan bahwa statusnya`in-use`. Buat log aliran VPC Amazon untuk antarmuka jaringan setiap instans, dan tinjau log aliran yang dihasilkan. Cari alamat IP broker saat Anda menjalankan `telnet` perintah, dan konfirmasikan paket koneksi`ACCEPTED`, termasuk paket pengembalian. Untuk informasi selengkapnya, dan untuk melihat contoh log aliran, lihat Contoh catatan log aliran di Panduan Pengembang Amazon VPC.

5. Jalankan `curl` perintah berikut untuk memeriksa konektivitas ke konsol web admin ActiveMQ.

```
$ curl https://b-1234a5b6-78cd-901e-2fgh-3i45j6k178l9-1.mq.us-
west-2.amazonaws.com:8162/index.html
```

Jika perintah berhasil, outputnya harus berupa dokumen HTML yang mirip dengan yang berikut ini.

```
<!DOCTYPE html PUBLIC "-//W3C//DTD HTML 4.01 Transitional//EN" "http://
www.w3.org/TR/html4/loose.dtd">
<html>
    <head>
        <meta http-equiv="Content-Type" content="text/html;
 charset=iso-8859-1" />
        <title>Apache ActiveMQ</title>
        ...
```

RabbitMQ on Amazon MQ

Untuk menguji RabbitMQ Anda di konektivitas jaringan broker Amazon MQ

1. Buka terminal baru atau jendela baris perintah.
2. Jalankan `nslookup` perintah berikut untuk menanyakan catatan DNS broker Anda. Ganti titik akhir dengan informasi Anda.

```
$ nslookup b-1234a5b6-78cd-901e-2fgh-3i45j6k178l9-1.mq.us-west-2.amazonaws.com
```

Jika kueri berhasil, Anda akan melihat output yang mirip dengan berikut ini.

```
Non-authoritative answer:
Server:  dns-resolver-corp-sfo-1.sfo.corp.amazon.com
Address:  172.10.123.456

Name:    rabbit-broker-1c23e456ca78-b9000123b4ebbab5.elb.us-
west-2.amazonaws.com
Addresses:  52.12.345.678
          52.23.234.56
          41.234.567.890
          54.123.45.678
Aliases:  b-1234a5b6-78cd-901e-2fgh-3i45j6k178l9-1.mq.us-west-2.amazonaws.com
```

3. Jalankan `telnet` perintah berikut untuk menguji jalur jaringan untuk broker Anda. Ganti titik akhir dengan informasi Anda. Anda dapat mengganti *`port`* dengan port 443 untuk konsol web, dan 5671 untuk menguji koneksi AMQP tingkat kabel.

```
$ telnet b-1234a5b6-78cd-901e-2fgh-3i45j6k178l9-1.mq.us-
west-2.amazonaws.com port
```

Jika perintah berhasil, Anda akan melihat output yang mirip dengan berikut ini.

```
Connected to b-1234a5b6-78cd-901e-2fgh-3i45j6k178l9-1.mq.us-
west-2.amazonaws.com.
Escape character is '^]'.
```

> **Note**
>
> Koneksi telnet akan menutup secara otomatis setelah beberapa detik.

4. Lakukan salah satu dari berikut ini.

   - Jika `telnet` perintah berhasil, periksa `ConnectionCount`metrik dan konfirmasikan bahwa broker belum mencapai nilai yang ditetapkan dalam kebijakan `max-connections`default. Anda juga dapat mengonfirmasi apakah batas telah tercapai dengan meninjau grup `Connection.log` log broker. Jika metrik ini lebih besar dari nol, setidaknya ada satu klien yang saat ini terhubung ke broker. Jika metrik menunjukkan koneksi nol, maka lakukan tes `telnet` jalur lagi. Anda mungkin perlu mengulangi proses ini jika koneksi ditutup sebelum broker Anda menerbitkan metrik koneksi baru. CloudWatch Metrik diterbitkan setiap menit.
   - Untuk broker tanpa aksesibilitas publik, jika `telnet` perintah gagal, periksa status antarmuka jaringan elastis broker Anda, dan konfirmasikan bahwa statusnya. `in-use` Buat log aliran VPC Amazon untuk setiap antarmuka jaringan, dan tinjau log aliran yang dihasilkan. Cari alamat IP pribadi broker saat Anda `telnet` perintah dipanggil, dan konfirmasikan paket koneksi`ACCEPTED`, termasuk paket pengembalian. Untuk informasi selengkapnya, dan untuk melihat contoh log aliran, lihat Contoh catatan log aliran di Panduan Pengembang Amazon VPC.

     > **Note**
     >
     > Langkah ini tidak berlaku untuk RabbitMQ di broker Amazon MQ dengan aksesibilitas publik.

5. Jalankan `curl` perintah berikut untuk memeriksa konektivitas ke konsol web admin RabbitMQ.

```
$ curl https://b-1234a5b6-78cd-901e-2fgh-3i45j6k178l9-1.mq.us-
west-2.amazonaws.com:443/index.html
```

Jika perintah berhasil, outputnya harus berupa dokumen HTML yang mirip dengan yang berikut ini.

```
<!DOCTYPE html>
```

```
<html>
    <head>
        <meta http-equiv="X-UA-Compatible" content="IE=edge" />
        <meta http-equiv="Content-Type" content="text/html; charset=utf-8" />
        <title>RabbitMQ Management</title>
        ...
```

Broker saya sedang berjalan, dan saya dapat memverifikasi konektivitas menggunakan**`telnet`**, tetapi klien saya tidak dapat terhubung dan mengembalikan pengecualian SSL.

Sertifikat titik akhir broker Anda mungkin telah diperbarui selama jendela pemeliharaan broker. Sertifikat broker Amazon MQ diputar secara berkala untuk memastikan ketersediaan dan keamanan broker yang berkelanjutan.

Sebaiknya gunakan otoritas sertifikat root Amazon (CA) di Amazon Trust Services untuk mengautentikasi di toko kepercayaan klien Anda. Semua sertifikat broker Amazon MQ ditandatangani dengan root CA ini. Dengan menggunakan root CA Amazon, Anda tidak perlu lagi mengunduh sertifikat broker Amazon MQ baru setiap kali ada pembaruan sertifikat di broker.

Saya membuat broker tetapi pembuatan broker gagal.

Jika broker Anda dalam `CREATION_FAILED` status, lakukan hal berikut.

- Periksa izin IAM Anda. Untuk membuat broker harus menggunakan kebijakan IAM AWS terkelola `AmazonMQFullAccess` atau memiliki kumpulan izin Amazon EC2 yang benar dalam kebijakan IAM kustom Anda. Untuk mempelajari lebih lanjut tentang izin Amazon EC2 yang diperlukan yang Anda perlukan, lihat Izin IAM yang diperlukan untuk membuat broker Amazon MQ.
- Periksa apakah subnet yang Anda pilih untuk broker Anda ada di Amazon Virtual Private Cloud (VPC) bersama. Untuk membuat broker Amazon MQ di VPC Amazon bersama, Anda harus membuatnya di akun yang memiliki VPC Amazon.

Broker saya memulai kembali dan saya tidak yakin mengapa.

Jika broker Anda telah memulai ulang secara otomatis, itu mungkin karena salah satu alasan berikut.

- Broker Anda mungkin telah memulai kembali karena jendela pemeliharaan mingguan yang dijadwalkan. Secara berkala, Amazon MQ melakukan pemeliharaan pada perangkat keras, sistem operasi, atau perangkat lunak mesin dari broker pesan. Durasi pemeliharaan bervariasi, tetapi dapat bertahan hingga dua jam, tergantung pada operasi yang dijadwalkan untuk broker pesan Anda. Broker dapat memulai kembali kapan saja selama jendela pemeliharaan dua jam. Untuk informasi lebih lanjut tentang jendela pemeliharaan broker, lihatthe section called “Penjadwalan pemeliharaan broker”.
- Jenis instans broker Anda mungkin tidak sesuai dengan beban kerja aplikasi Anda. Misalnya, menjalankan beban kerja produksi pada a `mq.t3.micro` dapat mengakibatkan broker kehabisan sumber daya. Pemanfaatan CPU yang tinggi, atau penggunaan memori broker yang tinggi dapat menyebabkan broker tiba-tiba memulai ulang. Untuk melihat berapa banyak CPU dan memori yang digunakan oleh broker Anda, gunakan CloudWatch metrik berikut untuk jenis mesin Anda.
  - ActiveMQ di Amazon MQ - `CpuUtilization` Periksa persentase unit komputasi Amazon EC2 yang dialokasikan yang saat ini digunakan broker. `HeapUsage`Periksa persentase batas memori ActiveMQ JVM yang digunakan broker saat ini.
  - RabbitMQ di Amazon MQ — Periksa `SystemCpuUtilization` persentase unit komputasi Amazon EC2 yang dialokasikan yang saat ini digunakan broker. Periksa `RabbitMQMemUsed` volume RAM yang digunakan dalam Bytes, dan `RabbitMQMemLimit` bagi dengan persentase memori yang digunakan oleh node RabbitMQ.

  Untuk informasi lebih lanjut tentang jenis instans broker dan cara memilih jenis instans yang tepat untuk beban kerja Anda, lihatBroker instance types.

Memecahkan masalah ActiveMQ di Amazon MQ

Gunakan informasi di bagian ini untuk membantu Anda mendiagnosis dan menyelesaikan masalah umum yang mungkin Anda temui saat bekerja dengan ActiveMQ di broker Amazon MQ.

Daftar Isi

- Saya tidak dapat melihat log umum atau audit untuk broker saya di CloudWatch Log meskipun saya telah mengaktifkan logging.
- Setelah broker restart atau jendela pemeliharaan, saya tidak dapat terhubung ke broker saya meskipun statusnyaRUNNING. Kenapa?
- Saya melihat beberapa klien saya terhubung ke broker, sementara yang lain tidak dapat terhubung.

- Saya melihat pengecualian org.apache.jasper.JasperException: An exception occurred processing JSP page pada konsol ActiveMQ saat melakukan operasi.

Saya tidak dapat melihat log umum atau audit untuk broker saya di CloudWatch Log meskipun saya telah mengaktifkan logging.

Jika Anda tidak dapat melihat log untuk broker Anda di CloudWatch Log, lakukan hal berikut.

1. Periksa apakah pengguna yang membuat atau me-reboot broker memiliki `logs:CreateLogGroup` izin. Jika Anda tidak menambahkan `CreateLogGroup` izin ke pengguna sebelum pengguna membuat atau me-reboot broker, Amazon MQ tidak akan membuat grup log.
2. Periksa apakah Anda telah mengonfigurasi kebijakan berbasis sumber daya untuk mengizinkan Amazon MQ memublikasikan log ke Log. CloudWatch Untuk mengizinkan Amazon MQ memublikasikan log ke grup CloudWatch log Log Anda, konfigurasikan kebijakan berbasis sumber daya untuk memberi Amazon MQ akses ke tindakan API Log berikut: CloudWatch
   - `CreateLogStream`— Membuat aliran CloudWatch log Log untuk grup log tertentu.
   - `PutLogEvents`— Mengirimkan peristiwa ke aliran CloudWatch log Log yang ditentukan.

Untuk informasi selengkapnya tentang mengonfigurasi ActiveMQ di Amazon MQ untuk memublikasikan CloudWatch log ke Log, lihat Mengonfigurasi logging.

Setelah broker restart atau jendela pemeliharaan, saya tidak dapat terhubung ke broker saya meskipun statusnya**RUNNING**. Kenapa?

Anda mungkin mengalami masalah koneksi setelah broker memulai ulang yang Anda mulai, setelah jendela pemeliharaan terjadwal selesai, atau dalam peristiwa kegagalan, di mana instance siaga diaktifkan. Dalam kedua kasus tersebut, masalah koneksi setelah broker restart kemungkinan besar disebabkan oleh sejumlah besar pesan yang bertahan di Amazon EFS broker Anda atau volume penyimpanan Amazon EBS. Selama restart, Amazon MQ memindahkan pesan tetap dari penyimpanan ke memori broker. Untuk mengonfirmasi diagnosis ini, Anda dapat memantau metrik berikut untuk Amazon MQ Anda CloudWatch untuk broker ActiveMQ:

- **`StoragePercentUsage`**Persentase besar pada atau mendekati 100 persen dapat menyebabkan broker menolak koneksi.

- **`JournalFilesForFullRecovery`**— Menunjukkan jumlah file jurnal yang akan diputar ulang setelah shutdown yang tidak bersih dan restart. Nilai yang meningkat, atau secara konsisten lebih tinggi dari satu, menunjukkan transaksi yang belum terselesaikan yang dapat menyebabkan masalah koneksi setelah restart.
- **`OpenTransactionCount`**— Angka yang lebih besar dari nol setelah restart menunjukkan bahwa broker akan mencoba menyimpan pesan yang dikonsumsi sebelumnya, sehingga menyebabkan masalah koneksi.

Untuk mengatasi masalah ini, kami sarankan untuk menyelesaikan transaksi XA Anda dengan a `rollback()` atau a. `commit()` Untuk informasi selengkapnya, dan untuk melihat contoh kode penyelesaian transaksi XA menggunakan`rollback()`, lihat memulihkan transaksi XA.

Saya melihat beberapa klien saya terhubung ke broker, sementara yang lain tidak dapat terhubung.

Jika broker Anda dalam `RUNNING` status dan beberapa klien dapat terhubung ke broker dengan sukses, sementara yang lain tidak dapat melakukannya, Anda mungkin telah mencapai batas koneksi tingkat kabel untuk broker. Untuk memverifikasi bahwa Anda telah mencapai batas koneksi tingkat kabel, lakukan hal berikut:

- Periksa log broker umum untuk ActiveMQ Anda di broker Amazon MQ di Log. CloudWatch Jika batas telah tercapai, Anda akan melihat `Reached Maximum Connections` di log broker. Untuk informasi lebih lanjut tentang CloudWatch Log untuk ActiveMQ di broker Amazon MQ, lihat. the section called “Memahami struktur logging di CloudWatch Log”

Setelah batas koneksi tingkat kabel tercapai, broker akan secara aktif menolak koneksi masuk tambahan. Untuk mengatasi masalah ini, kami sarankan untuk meningkatkan jenis instans broker Anda. Untuk informasi selengkapnya tentang memilih jenis instans terbaik untuk beban kerja Anda, lihatBroker instance types.

Jika Anda telah mengonfirmasi bahwa jumlah koneksi tingkat kabel Anda kurang dari batas koneksi broker, masalahnya mungkin terkait dengan me-reboot klien. Periksa log broker Anda untuk entri yang banyak dan sering. `... Inactive for longer than 600000 ms - removing ...` Entri log menunjukkan reboot klien atau masalah konektivitas. Efek ini lebih jelas ketika klien terhubung ke broker melalui Network Load Balancer (NLB) dengan klien yang sering memutuskan dan terhubung kembali ke broker. Ini lebih sering diamati pada klien berbasis kontainer.

Periksa log sisi klien Anda untuk detail lebih lanjut. Broker akan membersihkan koneksi TCP yang tidak aktif setelah 600000 ms, dan membebaskan soket koneksi.

Saya melihat pengecualian **`org.apache.jasper.JasperException: An exception occurred processing JSP page`** pada konsol ActiveMQ saat melakukan operasi.

Jika Anda menggunakan otentikasi sederhana dan mengonfigurasi `AuthorizationPlugin` untuk otorisasi antrian dan topik, pastikan untuk menggunakan `AuthorizationEntries` elemen dalam file konfigurasi XMLmu, dan izinkan izin `activemq-webconsole` grup untuk semua antrian dan topik. Ini memastikan bahwa konsol web ActiveMQ dapat berkomunikasi dengan broker ActiveMQ.

Contoh berikut `AuthorizationEntry` memberikan izin baca dan tulis untuk semua antrian dan topik ke grup. `activemq-webconsole`

```
<authorizationEntries>
    <authorizationEntry admin="activemq-webconsole,admins,users" topic=">"
 read="activemq-webconsole,admins,users" write="activemq-webconsole,admins,users" />
    <authorizationEntry admin="activemq-webconsole,admins,users" queue=">"
 read="activemq-webconsole,admins,users" write="activemq-webconsole,admins,users" />
</authorizationEntries>
```

Demikian pula ketika mengintegrasikan broker Anda dengan LDAP, pastikan untuk memberikan izin untuk grup. `amazonmq-console-admins` Untuk informasi lebih lanjut tentang integrasi LDAP, lihatthe section called “Cara kerja integrasi LDAP”.

Pemecahan masalah: RabbitMQ di Amazon MQ

Gunakan informasi di bagian ini untuk membantu Anda mendiagnosis dan menyelesaikan masalah umum yang mungkin Anda temui saat bekerja dengan RabbitMQ di broker Amazon MQ.

Daftar Isi

- Saya tidak dapat melihat metrik untuk antrian atau host virtual saya di. CloudWatch
- Bagaimana cara mengaktifkan plugin di RabbitMQ di Amazon MQ?
- Saya tidak dapat mengubah konfigurasi VPC Amazon untuk broker.
- Penerapan cluster telah menghentikan sinkronisasi antrian saya.

- Amazon MQ saya untuk broker instans tunggal RabbitMQ sedang dalam loop restart.
- Saya kehilangan akses ke semua akun administrator di broker saya.

Saya tidak dapat melihat metrik untuk antrian atau host virtual saya di. CloudWatch

Jika Anda tidak dapat melihat metrik untuk antrian atau host virtual CloudWatch, periksa apakah antrian atau nama host virtual berisi spasi kosong, tab, atau karakter non-ASCII lainnya.

Amazon MQ tidak dapat mempublikasikan metrik untuk host virtual dan antrian dengan nama yang berisi spasi kosong, tab, atau karakter non-ASCII lainnya.

Untuk informasi selengkapnya tentang nama dimensi, lihat Dimensi di Referensi Amazon CloudWatch API.

Bagaimana cara mengaktifkan plugin di RabbitMQ di Amazon MQ?

RabbitMQ di Amazon MQ saat ini hanya mendukung manajemen RabbitMQ, sekop, federasi, plugin pertukaran hash konsisten, yang diaktifkan secara default. Untuk informasi selengkapnya tentang penggunaan plugin yang didukung, lihatthe section called “Plugin”.

Saya tidak dapat mengubah konfigurasi VPC Amazon untuk broker.

Amazon MQ tidak mendukung perubahan konfigurasi VPC Amazon setelah broker Anda dibuat. Harap dicatat bahwa Anda perlu membuat broker baru dengan konfigurasi VPC Amazon baru dan memperbarui URL koneksi klien dengan URL koneksi broker baru.

Penerapan cluster telah menghentikan sinkronisasi antrian saya.

Saat menangani alarm memori tinggi RabbitMQ, Anda mungkin menemukan bahwa pesan pada satu atau beberapa antrian tidak dapat dikonsumsi. Antrian ini mungkin dalam proses sinkronisasi pesan antar node, di mana antrian masing-masing menjadi tidak tersedia untuk diterbitkan dan dikonsumsi. Sinkronisasi antrian mungkin menjadi berhenti karena alarm memori yang tinggi, dan bahkan berkontribusi pada alarm memori.

Untuk informasi tentang menghentikan dan mencoba kembali sinkronisasi antrian yang dijeda, lihat. the section called “Menyelesaikan sinkronisasi antrean yang dijeda”

Amazon MQ saya untuk broker instans tunggal RabbitMQ sedang dalam loop restart.

Amazon MQ untuk broker instans tunggal RabbitMQ yang memunculkan alarm memori tinggi berisiko menjadi tidak tersedia jika restart dan tidak memiliki cukup memori untuk memulai. Hal ini dapat menyebabkan RabbitMQ memasuki loop restart dan mencegah interaksi lebih lanjut dengan broker sampai masalah teratasi. Jika broker Anda dalam loop restart, Anda tidak akan dapat menerapkan praktik terbaik yang direkomendasikan Amazon MQ untuk menyelesaikan alarm memori tinggi.

Untuk memulihkan broker Anda, kami sarankan untuk meningkatkan ke jenis instans yang lebih besar dengan lebih banyak memori. Tidak seperti dalam penerapan cluster, Anda dapat memutakhirkan broker instans tunggal saat mengalami alarm memori tinggi karena tidak ada sinkronisasi antrian untuk dilakukan antar node selama restart.

Saya kehilangan akses ke semua akun administrator di broker saya.

Anda dapat memulihkan akses menggunakan otentikasi IAM. Aktifkan federasi identitas web keluar untuk AWS akun Anda, buat peran IAM dengan izin untuk mendapatkan token identitas web, konfigurasikan broker Anda untuk menerima otentikasi IAM melalui OAuth 2.0, lalu gunakan kredensyal IAM untuk mendapatkan token JWT dan membuat pengguna administrator baru. Untuk petunjuk mendetail, lihat the section called “Menggunakan otentikasi dan otorisasi IAM”.

ActiveMQ di Amazon MQ: Alarm Antarmuka Jaringan Elastis yang Dihapus

ActiveMQ di Amazon MQ akan menaikkan alarm BROKER_ENI_DELETED saat Anda menghapus Antarmuka Jaringan Elastis (ENI) broker. Ketika Anda pertama kali membuat broker Amazon MQ, Amazon MQ menyediakan antarmuka jaringan elastis pada Virtual Private Cloud (VPC) di bawah akun Anda dan memerlukan sejumlah izin EC2.

Anda tidak harus memodifikasi atau menghapus antarmuka jaringan ini. Memodifikasi atau menghapus antarmuka jaringan dapat menyebabkan koneksi hilang permanen antara VPC dan broker Anda. Jika Anda ingin menghapus antarmuka jaringan, Anda harus menghapus broker terlebih dahulu.

ActiveMQ di Amazon MQ: Broker Kehabisan Alarm Memori

ActiveMQ di Amazon MQ akan menaikkan alarm BROKER_OOM ketika broker mengalami loop restart karena kapasitas memori yang tidak mencukupi. Ketika broker berada dalam loop restart, juga disebut loop bouncing, broker memulai upaya pemulihan berulang dalam jangka waktu singkat. Pialang yang tidak dapat menyelesaikan start-up karena kapasitas memori yang tidak mencukupi dapat memasuki loop restart, di mana interaksi dengan broker terbatas.

Amazon MQ memungkinkan metrik untuk broker Anda secara default. Anda dapat melihat metrik broker Anda dengan mengakses CloudWatch konsol Amazon, atau dengan menggunakan API. CloudWatch Metrik berikut berguna saat mendiagnosis alarm ActiveMQ BROKER_OOM:

Metrik Amazon MQ CloudWatch	Alasan penggunaan memori tinggi	
`TotalMessageCount`	Pesan disimpan dalam memori sampai dikonsumsi atau dibuang. Jumlah pesan yang tinggi mungkin menunjukkan pemanfaatan sumber daya yang berlebihan dan dapat menyebabkan alarm memori yang tinggi.	
`HeapUsage`	Persentase batas memori ActiveMQ JVM yang saat ini digunakan oleh broker. Persentase yang lebih tinggi menunjukkan broker menggunakan sumber daya yang signifikan dan dapat menyebabkan alarm OOM.	
`ConnectionCount`	Koneksi klien menggunakan memori, dan terlalu banyak koneksi simultan dapat	

Metrik Amazon MQ CloudWatch	Alasan penggunaan memori tinggi	
	menyebabkan alarm memori tinggi.	
`CpuUtilization`	Persentase unit komputasi EC2 yang dialokasikan yang saat ini digunakan broker.	
`TotalConsumerCount`	Untuk setiap konsumen yang terhubung ke broker, sejumlah pesan dimuat dari penyimpan an ke memori sebelum dikirim ke konsumen. Sejumlah besar koneksi konsumen dapat menyebabkan penggunaa n memori yang tinggi dan menyebabkan alarm memori yang tinggi.	

Untuk mencegah restart loop dan menghindari alarm BROKER_OOM, pastikan pesan dikonsumsi dengan cepat. Anda dapat melakukan ini dengan memilih jenis instans broker yang paling efektif, dan juga membersihkan Antrian Surat Mati Anda untuk membuang pesan yang tidak terkirim atau kedaluwarsa. Anda dapat mempelajari lebih lanjut tentang memastikan kinerja yang efektif di ActiveMQ di praktik terbaik Amazon MQ.

Amazon MQ untuk RabbitMQ: Alarm memori tinggi

Amazon MQ untuk RabbitMQ akan meningkatkan alarm memori tinggi ketika penggunaan memori broker, diidentifikasi dengan CloudWatch metrik, melebihi batas memori`RabbitMQMemUsed`, diidentifikasi oleh. `RabbitMQMemLimit`

Broker RabbitMQ yang telah menaikkan alarm memori tinggi akan memblokir semua klien yang menerbitkan pesan. Broker Anda dapat memasuki loop restart, mengalami sinkronisasi antrian yang dijeda, atau mengembangkan masalah lain yang mempersulit diagnosis dan resolusi alarm.

Untuk mendiagnosis dan mengatasi alarm memori tinggi, pertama-tama ikuti semua praktik terbaik untuk RabbitMQ, lalu selesaikan langkah-langkah berikut.

> **Important**
>
> - `RabbitMQMemLimit`diatur oleh Amazon MQ dan secara khusus disetel mengingat memori yang tersedia untuk setiap jenis instance host.
> - Amazon MQ tidak akan me-restart broker yang mengalami alarm memori tinggi dan akan mengembalikan pengecualian untuk operasi `RebootBroker`API selama broker terus menaikkan alarm.

Langkah 1: Diagnosis alarm memori tinggi

Ada dua cara untuk mendiagnosis alarm memori tinggi di Amazon MQ Anda untuk broker RabbitMQ. Kami menyarankan Anda memeriksa konsol web RabbitMQ dan metrik Amazon MQ di. CloudWatch

Mendiagnosis alarm memori tinggi menggunakan konsol web RabbitMQ

Konsol web RabbitMQ dapat menghasilkan dan menampilkan informasi penggunaan memori terperinci untuk setiap node. Anda dapat menemukan informasi ini dengan melakukan hal berikut:

1. Masuk Konsol Manajemen AWS dan buka konsol web RabbitMQ broker Anda.
2. Pada konsol RabbitMQ, pada halaman Ikhtisar, pilih nama node dari daftar Nodes.
3. Pada halaman detail node, pilih Detail memori untuk memperluas bagian untuk melihat informasi penggunaan memori node.

Informasi penggunaan memori yang disediakan RabbitMQ di konsol web dapat membantu Anda menentukan sumber daya mana yang mungkin menghabiskan terlalu banyak memori dan berkontribusi pada alarm memori tinggi. Untuk informasi selengkapnya tentang detail penggunaan memori yang tersedia melalui konsol web RabbitMQ, lihat Penalaran Tentang Penggunaan Memori di situs web Dokumentasi Server RabbitMQ.

Mendiagnosis alarm memori tinggi menggunakan metrik Amazon MQ

Amazon MQ memungkinkan metrik untuk broker Anda secara default. Anda dapat melihat metrik broker Anda dengan mengakses CloudWatch konsol, atau dengan menggunakan API. CloudWatch Metrik berikut berguna saat mendiagnosis alarm memori tinggi RabbitMQ.

Metrik Amazon MQ CloudWatch	Alasan penggunaan memori tinggi	
`MessageCount`	Pesan disimpan dalam memori sampai dikonsumsi atau dibuang. Jumlah pesan yang tinggi mungkin menunjukkan pemanfaatan sumber daya yang berlebihan dan dapat menyebabkan alarm memori yang tinggi.	
`QueueCount`	Antrian disimpan dalam memori, dan sejumlah besar antrian dapat menyebabkan alarm memori yang tinggi.	
`ConnectionCount`	Koneksi klien menggunakan memori, dan terlalu banyak koneksi simultan dapat menyebabkan alarm memori tinggi.	
`ChannelCount`	Mirip dengan koneksi, saluran yang dibuat menggunakan setiap koneksi juga disimpan dalam memori node, dan sejumlah besar saluran dapat menyebabkan alarm memori tinggi.	
`ConsumerCount`	Untuk setiap konsumen yang terhubung ke broker, sejumlah pesan dimuat dari penyimpanan ke memori sebelum dikirim ke konsumen. Sejumlah besar koneksi konsumen dapat	

Metrik Amazon MQ CloudWatch	Alasan penggunaan memori tinggi	
	menyebabkan penggunaan memori yang tinggi dan menyebabkan alarm memori yang tinggi.	
`PublishRate`	Menerbitkan pesan menggunakan memori broker. Jika tingkat di mana pesan dipublikasikan ke broker terlalu tinggi dan secara signifikan melebihi tingkat di mana broker mengirimkan pesan kepada konsumen, broker mungkin menaikkan alarm memori yang tinggi.	

Langkah 2: Alamat dan cegah alarm memori tinggi

> **Note**
>
> Mungkin diperlukan waktu hingga beberapa jam agar status RABBITMQ_MEMORY_ALARM dihapus setelah Anda mengambil tindakan yang diperlukan.

Ikuti semua praktik terbaik untuk RabbitMQ sebagai metode pencegahan umum. Untuk setiap kontributor spesifik yang Anda identifikasi, kami merekomendasikan serangkaian tindakan berikut untuk mengatasi dan mencegah alarm memori tinggi RabbitMQ.

Sumber penggunaan memori tinggi	Rekomendasi Amazon MQ untuk menangani	Rekomendasi Amazon MQ untuk mencegah
Jumlah pesan	Konsumsi pesan yang dipublikasikan ke antrian, bersihkan pesan dari antrian,	Aktifkan antrian malas, dan atur atau kurangi batas kedalaman antrian.

Sumber penggunaan memori tinggi	Rekomendasi Amazon MQ untuk menangani	Rekomendasi Amazon MQ untuk mencegah
	atau hapus antrian dari broker Anda.	
Jumlah antrian	Kurangi jumlah antrian.	Atur atau kurangi batas hitungan antrian.
Jumlah koneksi	Kurangi jumlah koneksi.	Atur atau kurangi batas jumlah koneksi.
Jumlah saluran	Kurangi jumlah saluran.	Tetapkan jumlah maksimum saluran per koneksi pada aplikasi klien.
Jumlah konsumen	Kurangi jumlah konsumen yang terhubung ke broker.	Tetapkan batas pra-pengambilan konsumen yang kecil.
Tingkat penerbitan pesan	Kurangi tingkat di mana penerbit mengirim pesan ke broker.	Aktifkan konfirmasi penerbit.
Tingkat upaya koneksi klien	Kurangi frekuensi di mana klien mencoba untuk terhubung ke broker untuk mempublikasikan atau mengkonsumsi pesan, atau mengkonfigurasi broker.	Gunakan koneksi yang berumur lebih lama untuk mengurangi jumlah dan frekuensi upaya koneksi.

Setelah alarm memori broker Anda teratasi, Anda dapat meningkatkan jenis instans host Anda ke instance dengan sumber daya tambahan. Untuk informasi tentang cara memperbarui jenis instans broker Anda, lihat `UpdateBrokerInput`di Referensi API REST Amazon MQ.

> **Note**
>
> Anda tidak dapat menurunkan versi broker dari tipe `mq.m5.x` instans ke tipe `mq.t3.micro` instans. Untuk downgrade, Anda harus menghapus broker Anda dan membuat yang baru.

RabbitMQ di Amazon MQ: Kunci Tidak Valid AWS Key Management Service

RabbitMQ di Amazon MQ akan memunculkan kode yang diperlukan tindakan kritis INVALID_KMS_KEY saat broker yang dibuat dengan pelanggan yang dikelola AWS KMS key(CMK) mendeteksi bahwa kunci (KMS) dinonaktifkan. AWS Key Management Service Broker RabbitMQ dengan CMK secara berkala memverifikasi bahwa kunci KMS diaktifkan dan broker memiliki semua hibah yang diperlukan. Jika RabbitMQ tidak dapat memverifikasi bahwa kunci diaktifkan, broker dikarantina dan RabbitMQ akan mengembalikan INVALID_KMS_KEY.

Tanpa kunci KMS aktif, broker tidak memiliki izin dasar untuk kunci KMS yang dikelola pelanggan. Broker tidak dapat melakukan operasi kriptografi menggunakan kunci Anda sampai Anda mengaktifkan kembali kunci Anda dan broker memulai ulang. Pialang RabbitMQ dengan kunci KMS yang dinonaktifkan dikarantina untuk mencegah kerusakan. Setelah RabbitMQ menentukan kunci KMS aktif kembali, broker Anda dihapus dari karantina. Amazon MQ tidak memulai ulang broker dengan kunci KMS yang dinonaktifkan dan mengembalikan pengecualian untuk operasi `RebootBroker` API selama broker terus memiliki kunci KMS yang tidak valid.

Mendiagnosis dan menangani INVALID_KMS_KEY

Untuk mendiagnosis dan mengatasi kode yang diperlukan tindakan INVALID_KMS_KEY, Anda harus menggunakan Command AWS Line Interface (CLI) dan konsol. AWS Key Management Service

Untuk mengaktifkan kembali kunci KMS Anda

1. Panggil `DescribeBroker` metode untuk mengambil untuk broker CMK Anda. `kmsKeyId`
2. Masuk ke AWS Key Management Service konsol.
3. Pada halaman kunci yang dikelola Pelanggan, cari ID Kunci KMS dari broker yang bermasalah dan verifikasi statusnya Diaktifkan.
4. Jika kunci KMS Anda telah dinonaktifkan, aktifkan kembali kunci dengan memilih Tindakan Kunci, lalu pilih Aktifkan. Setelah kunci Anda diaktifkan kembali, Anda harus menunggu RabbitMQ menghapus broker dari karantina.

Untuk memverifikasi bahwa hibah yang diperlukan masih terkait dengan kunci KMS broker, hubungi `ListGrant` ListGrant metode untuk memverifikasi itu `mq_rabbit_grant` dan `mq_grant` ada. Jika hibah atau kunci KMS telah dihapus, Anda harus menghapus broker dan membuat yang baru dengan semua hibah yang diperlukan. Untuk langkah-langkah menghapus broker, lihat Menghapus broker.

Untuk mencegah kode yang diperlukan tindakan kritis INVALID_KMS_KEY, jangan menghapus atau menonaktifkan kunci KMS atau hibah CMK secara manual. Jika Anda ingin menghapus kunci, hapus broker terlebih dahulu.

RabbitMQ di Amazon MQ: Alarm batas disk

Alarm batas disk adalah indikasi bahwa volume disk yang digunakan oleh node RabbitMQ telah menurun karena tingginya jumlah pesan yang tidak dikonsumsi saat pesan baru ditambahkan. RabbitMQ akan menaikkan alarm batas disk ketika ruang disk kosong broker, yang diidentifikasi oleh CloudWatch metrik Amazon`RabbitMQDiskFree`, mencapai batas disk, diidentifikasi oleh. `RabbitMQDiskFreeLimit` `RabbitMQDiskFreeLimit`ditetapkan oleh Amazon MQ dan telah ditentukan dengan mempertimbangkan ruang disk yang tersedia untuk setiap jenis instans broker.

Broker RabbitMQ di Amazon MQ yang telah menaikkan alarm batas disk akan menjadi tidak tersedia untuk pesan baru yang diterbitkan. Jika Anda memiliki penerbit dan konsumen pada koneksi yang sama, konsumen juga tidak akan tersedia untuk menerima pesan. Saat menjalankan RabbitMQ dalam sebuah cluster, alarm disk berada di seluruh cluster. Jika satu node berada di bawah batas, semua node lain akan memblokir pesan yang masuk. Karena kurangnya ruang disk, broker Anda mungkin juga mengalami masalah lain yang mempersulit diagnosis dan resolusi alarm.

Amazon MQ tidak akan me-restart broker yang mengalami alarm disk dan akan mengembalikan pengecualian untuk operasi `RebootBroker` API selama broker terus menaikkan alarm.

> **Note**
>
> Anda tidak dapat menurunkan versi broker dari tipe `mq.m5` instans ke tipe `mq.t3.micro` instans. Jika Anda ingin downgrade, Anda harus menghapus broker Anda dan membuat yang baru.

Mendiagnosis dan menangani alarm batas disk

Amazon MQ memungkinkan metrik untuk broker Anda secara default. Anda dapat melihat metrik broker Anda dengan mengakses CloudWatch konsol Amazon, atau dengan menggunakan API. CloudWatch `MessageCount`adalah metrik yang berguna saat mendiagnosis alarm batas disk RabbitMQ. Pesan disimpan dalam memori sampai dikonsumsi atau dibuang. Jumlah pesan yang tinggi menunjukkan pemanfaatan penyimpanan disk yang berlebihan dan dapat menyebabkan alarm disk.

Untuk mendiagnosis alarm batas disk, gunakan Amazon MQ Management Console untuk:

- Buat koneksi baru untuk mengkonsumsi pesan yang dipublikasikan ke antrian.
- Bersihkan pesan dari antrian.
- Hapus antrian dari broker Anda.

> **Note**
>
> Mungkin diperlukan waktu hingga beberapa jam agar status RABBITMQ_DISK_ALARM dihapus setelah Anda mengambil tindakan yang diperlukan.

Untuk mencegah alarm batas disk berulang, Anda dapat memutakhirkan jenis instans host ke instance dengan sumber daya tambahan. Untuk informasi tentang cara memperbarui jenis instans broker Anda, lihat `UpdateBrokerInput` di Referensi API REST Amazon MQ. Kami juga menyarankan agar penerbit dan konsumen Anda tetap pada koneksi yang berbeda.

Amazon MQ untuk RabbitMQ: Alarm perubahan jenis instans

`RABBITMQ_CLUSTER_DISK_USAGE_TOO_HIGH_FOR_INSTANCE_CHANGE`adalah indikasi bahwa perubahan jenis instans broker yang diminta tidak dapat dilanjutkan karena penggunaan disk yang tinggi pada node RabbitMQ saat ini. Amazon MQ untuk RabbitMQ akan memunculkan alarm ini ketika ketika penggunaan disk saat ini melebihi apa yang akan tersedia pada jenis instance yang diminta, seperti yang diidentifikasi oleh metrik. CloudWatch `RabbitMQDiskFree`

Broker RabbitMQ yang memasuki `RABBITMQ_CLUSTER_DISK_USAGE_TOO_HIGH_FOR_INSTANCE_CHANGE` status akan terus tersedia untuk aplikasi Anda, tetapi perubahan jenis instance yang diminta tidak akan dilanjutkan. Amazon MQ memungkinkan broker memulai ulang dalam status ini, tetapi Anda tidak dapat mengubah jenis instans saat penggunaan disk tetap di atas ambang batas untuk jenis instans yang diminta. Broker akan mengembalikan pengecualian untuk operasi `ModifyBroker` API yang mencoba mengubah jenis instance saat berada dalam status ini.

Mendiagnosis dan menangani alarm perubahan tipe instance

Amazon MQ memungkinkan metrik untuk broker Anda secara default. Anda dapat melihat metrik broker Anda dengan mengakses CloudWatch konsol atau dengan menggunakan

API. CloudWatch `MessageCount`dan `RabbitMQDiskFree` metrik dapat digunakan untuk mendiagnosis`RABBITMQ_CLUSTER_DISK_USAGE_TOO_HIGH_FOR_INSTANCE_CHANGE`.

Untuk menyelesaikan status karantina dan mengizinkan perubahan jenis instans untuk melanjutkan, gunakan Amazon MQ Management Console untuk:

- Buat koneksi baru untuk mengkonsumsi pesan yang dipublikasikan ke antrian.
- Bersihkan pesan dari antrian.
- Hapus antrian dari broker Anda.

> **Note**
>
> Mungkin diperlukan waktu hingga beberapa jam agar `RABBITMQ_CLUSTER_DISK_USAGE_TOO_HIGH_FOR_INSTANCE_CHANGE` status dihapus setelah Anda mengambil tindakan yang diperlukan.

RabbitMQ di Amazon MQ: IAM Asumsikan Peran Tidak Valid

RabbitMQ di Amazon MQ akan memunculkan kode yang diperlukan tindakan kritis INVALID_ASSUMEROLE saat peran IAM yang ditentukan ARN tidak valid atau tidak dapat diasumsikan oleh Amazon MQ. `aws.arns.assume_role_arn` Ini dapat terjadi ketika peran tidak ada, berada di AWS akun yang berbeda dari broker, atau tidak memiliki hubungan kepercayaan yang diperlukan dengan mq.amazonaws.com.

Broker di karantina RABBITMQ_INVALID_ASSUMEROLE tidak dapat mengambil kredensyal atau sertifikat yang diperlukan untuk otentikasi LDAP, sehingga otentikasi LDAP tidak tersedia. Jika LDAP adalah satu-satunya metode otentikasi yang dikonfigurasi, pengguna tidak akan dapat terhubung ke broker. Peran IAM diperlukan oleh Amazon MQ untuk AWS mengakses sumber daya yang dirujuk ARNs oleh dalam konfigurasi broker, AWS Secrets Manager seperti rahasia atau objek Amazon S3 yang digunakan untuk otentikasi LDAP.

Mendiagnosis dan menangani RABBITMQ_INVALID_ASSUMEROLE

Untuk mendiagnosis dan menangani kode yang diperlukan tindakan RABBITMQ_INVALID_ASSUMEROLE, Anda harus menggunakan Amazon Logs dan konsol. CloudWatch AWS Identity and Access Management

Untuk mengatasi masalah peran asumsi yang tidak valid

1. Arahkan ke Amazon CloudWatch Logs Insights dan jalankan kueri berikut terhadap grup `/aws/amazonmq/broker/<broker-id>/general` log broker Anda:

```
fields @timestamp, @message
| sort @timestamp desc
| filter @message like /error.*aws_arn_config/
| limit 10000
```

2. Cari pesan kesalahan yang mirip dengan:

```
[error] <0.254.0> aws_arn_config: {handle_assume_role,{error,
{assume_role_failed,"AWS service is unavailable"}}}
```

3. Periksa konfigurasi peran IAM dan perbaiki masalah apa pun seperti:

   - Pastikan peran ada di AWS akun yang sama dengan broker
   - Verifikasi kebijakan kepercayaan memungkinkan mq.amazonaws.com untuk mengambil peran
   - Konfirmasikan bahwa peran memiliki izin yang sesuai untuk mengakses sumber daya yang diperlukan AWS

4. Validasi perbaikan menggunakan titik akhir API validasi akses ARN sebelum memperbarui konfigurasi broker.

5. Perbarui konfigurasi broker dan reboot broker.

RabbitMQ di Amazon MQ: LDAP ARN tidak valid

RabbitMQ di Amazon MQ akan memunculkan kode yang diperlukan tindakan kritis INVALID_ARN_LDAP saat ARN yang dikonfigurasi untuk kata sandi akun layanan LDAP tidak valid atau tidak dapat diakses. Ini berlaku untuk ARNs yang ditentukan dalam `aws.arns.auth_ldap.dn_lookup_bind.password` atau`aws.arns.auth_ldap.other_bind.password`, yang harus mereferensikan AWS Secrets Manager rahasia yang berisi kata sandi teks biasa.

Broker di karantina RABBITMQ_INVALID_ARN_LDAP tidak dapat mengautentikasi dengan akun layanan LDAP, membuat otentikasi LDAP tidak tersedia. Jika LDAP adalah satu-satunya metode otentikasi yang dikonfigurasi, pengguna tidak akan dapat terhubung ke broker. Tidak valid ARNs dapat disebabkan oleh sintaks ARN yang salah bentuk, referensi ke rahasia yang tidak ada, rahasia yang terletak di wilayah yang AWS berbeda dari broker, atau secretsmanager yang tidak mencukupi: izin dalam peran IAM. GetSecretValue

Mendiagnosis dan menangani RABBITMQ_INVALID_ARN_LDAP

Untuk mendiagnosis dan menangani kode yang diperlukan tindakan RABBITMQ_INVALID_ARN_LDAP, Anda harus menggunakan Amazon Logs dan konsol. CloudWatch

Untuk mengatasi masalah ARN LDAP yang tidak valid

1. Arahkan ke Amazon CloudWatch Logs Insights dan jalankan kueri berikut terhadap grup `/aws/amazonmq/broker/<broker-id>/general` log broker Anda:

```
fields @timestamp, @message
| sort @timestamp desc
| filter @message like /error.*aws_arn_config/
| limit 10000
```

2. Cari pesan kesalahan yang mirip dengan:

```
[error] <0.254.0> aws_arn_config: {<<"could not resolve
 ARN 'arn:aws:secretsmanager:xxx' for configuration
 'aws.arns.auth_ldap.dn_lookup_bind.password', error: \"AWS service is unavailable
\"">>,{error,"AWS service is unavailable"}}
```

3. Periksa rahasia Secrets Manager dan perbaiki masalah apa pun seperti:

   - Verifikasi rahasia ada di AWS wilayah yang sama dengan broker
   - Konfirmasikan sintaks ARN benar
   - Pastikan peran IAM memiliki izin secretsmanager: GetSecretValue

4. Validasi perbaikan menggunakan titik akhir API validasi akses ARN sebelum memperbarui konfigurasi broker.
5. Perbarui konfigurasi broker dan reboot broker.

RabbitMQ di Amazon MQ: HTTP ARN tidak valid

RabbitMQ di Amazon MQ akan memunculkan kode tindakan kritis INVALID_ARN_HTTP yang diperlukan ketika satu atau ARNs beberapa sertifikat SSL atau file kunci untuk HTTP auth_backend tidak valid atau tidak dapat diakses. Ini berlaku untuk ARNS yang ditentukan dalam`aws.arns.auth_http.ssl_options.cacertfile`, `aws.arns.auth_http.ssl_options.certfile` atau`aws.arns.auth_http.ssl_options.keyfile`, yang harus mereferensikan objek AWS Secrets Manager dan rahasia Amazon S3 yang berisi sertifikat dan kunci pribadi.

Broker di karantina RABBITMQ_INVALID_ARN_HTTP tidak dapat mengautentikasi melalui server HTTP. Jika HTTP adalah satu-satunya metode otentikasi yang dikonfigurasi, pengguna tidak akan dapat terhubung ke broker. Tidak valid ARNs dapat disebabkan oleh sintaks ARN yang salah bentuk, referensi ke rahasia yang tidak ada, rahasia yang terletak di wilayah yang AWS berbeda dari broker, atau izin s3GetObject: /secretsmanager: yang tidak mencukupi dalam peran IAM. GetSecretValue

Mendiagnosis dan menangani RABBITMQ_INVALID_ARN_HTTP

Untuk mendiagnosis dan menangani kode yang diperlukan tindakan RABBITMQ_INVALID_ARN_HTTP, Anda harus menggunakan Amazon Logs dan konsol. CloudWatch

Untuk mengatasi masalah ARN HTTP yang tidak valid

1. Arahkan ke Amazon CloudWatch Logs Insights dan jalankan kueri berikut terhadap grup `/aws/amazonmq/broker/<broker-id>/general` log broker Anda:

```
fields @timestamp, @message
| sort @timestamp desc
| filter @message like /error.*aws_arn_config/
| limit 10000
```

2. Cari pesan kesalahan yang mirip dengan:

```
[error] <0.209.0> aws_arn_config: {<<"could not resolve ARN 'arn:aws:s3:::xxxx' for
 configuration 'aws.arns.auth_http.ssl_options.certfile', error: \"AWS service is
 unavailable\"">>,{error,"AWS service is unavailable"}}
```

3. Periksa rahasia Object/Secrets Manajer S3 dan perbaiki masalah apa pun seperti:
   - Verifikasi sumber daya yang ada di AWS wilayah yang sama dengan broker
   - Konfirmasikan sintaks ARN benar
   - Pastikan peran IAM memiliki izin s3: GetObject dan secretsmanager: GetSecretValue
4. Validasi perbaikan menggunakan titik akhir API validasi akses ARN sebelum memperbarui konfigurasi broker.
5. Perbarui konfigurasi broker dan reboot broker.

RabbitMQ di Amazon MQ: SSL ARN tidak valid

RabbitMQ di Amazon MQ akan memunculkan kode yang diperlukan tindakan kritis INVALID_ARN_SSL ketika satu atau beberapa truststore sertifikat CA untuk auth_mechanism EKSTERNAL tidak valid atau tidak dapat diakses. ARNs Ini berlaku untuk ARNS yang ditentukan dalam `aws.arns.ssl_options.cacertfile` atau`aws.arns.management.ssl.cacertfile`, yang harus mereferensikan objek Amazon S3 atau ACM PCA yang berisi sertifikat.

Pialang di karantina RABBITMQ_INVALID_ARN_SSL tidak dapat mengautentikasi sertifikat klien selama jabat tangan TLS bersama karena tidak ada truststore yang valid yang dikonfigurasi. Jika mekanisme autentikasi EKSTERNAL adalah satu-satunya metode otentikasi yang dikonfigurasi, pengguna tidak akan dapat terhubung ke broker. Tidak valid ARNs dapat disebabkan oleh sintaks ARN yang salah bentuk, referensi ke objek S3 yang tidak ada, objek S3 yang terletak di wilayah yang AWS berbeda dari broker, atau izin s3: /acm-pca: yang tidak mencukupi dalam peran IAM. GetObject GetCertificateAuthorityCertificate

Mendiagnosis dan menangani RABBITMQ_INVALID_ARN_SSL

Untuk mendiagnosis dan menangani kode yang diperlukan tindakan RABBITMQ_INVALID_ARN_SSL, Anda harus menggunakan Amazon Logs dan konsol. CloudWatch

Untuk mengatasi masalah SSL ARN yang tidak valid

1. Arahkan ke Amazon CloudWatch Logs Insights dan jalankan kueri berikut terhadap grup `/aws/amazonmq/broker/<broker-id>/general` log broker Anda:

```
fields @timestamp, @message
| sort @timestamp desc
| filter @message like /error.*aws_arn_config/
| limit 10000
```

2. Cari pesan kesalahan yang mirip dengan:

```
[error] <0.209.0> aws_arn_config: {<<"could not resolve ARN 'arn:aws:acm-pca:xxxx'
 for configuration 'aws.arns.ssl_options.cacertfile', error: \"AWS service is
 unavailable\"">>,{error,"AWS service is unavailable"}}
```

3. Periksa Objek S3/ACM-PCA dan perbaiki masalah apa pun seperti:

   - Verifikasi rahasia ada di AWS wilayah yang sama dengan broker
   - Konfirmasikan sintaks ARN benar
   - Pastikan peran IAM memiliki izin s3: GetObject /acm-pca: GetCertificateAuthorityCertificate

4. Validasi perbaikan menggunakan titik akhir API validasi akses ARN sebelum memperbarui konfigurasi broker.
5. Perbarui konfigurasi broker dan reboot broker.

RabbitMQ di Amazon MQ: ARN tidak valid

RabbitMQ di Amazon MQ akan memunculkan kode tindakan kritis INVALID_ARN yang diperlukan ketika satu atau lebih yang dikonfigurasi di broker tidak valid atau tidak dapat diakses. ARNs Ini berlaku untuk ARNs digunakan untuk sertifikat SSL, AWS Secrets Manager rahasia, objek Amazon S3, atau referensi sumber daya AWS lainnya yang tidak tercakup oleh kode karantina yang lebih spesifik seperti RABBITMQ_INVALID_ARN_LDAP atau RABBITMQ_INVALID_ASSUMEROLE.

Pialang di karantina RABBITMQ_INVALID_ARN mungkin mengalami fungsionalitas yang terdegradasi tergantung pada mana yang tidak valid. ARNs Fitur yang bergantung pada sumber

daya yang tidak dapat diakses tidak akan tersedia, dan broker akan mencatat kesalahan yang menunjukkan ARN mana yang gagal diselesaikan. Dampak pada ketersediaan broker tergantung pada apakah ARN yang tidak valid diperlukan untuk operasi broker penting.

Mendiagnosis dan menangani RABBITMQ_INVALID_ARN

Untuk mendiagnosis dan menangani kode yang diperlukan tindakan RABBITMQ_INVALID_ARN, Anda harus menggunakan CloudWatch Amazon Logs dan konsol layanan yang sesuai untuk sumber daya yang terpengaruh. AWS

Untuk mengatasi masalah ARN yang tidak valid

1. Arahkan ke Amazon CloudWatch Logs Insights dan jalankan kueri berikut terhadap grup `/aws/amazonmq/broker/<broker-id>/general` log broker Anda:

```
fields @timestamp, @message
| sort @timestamp desc
| filter @message like /error.*aws_arn_config/
| limit 10000
```

2. Cari pesan kesalahan yang mirip dengan:

```
[error] <0.254.0> aws_arn_config: {<<"could not resolve ARN
 'arn:aws:s3:::bucket-name/certificate.pem' for configuration
 'aws.arns.auth_ldap.ssl_options.cacertfile', error: \"AWS service is unavailable
\"">>,{error,"AWS service is unavailable"}}
```

3. Periksa AWS sumber daya dan perbaiki masalah apa pun seperti:

   - Verifikasi sumber daya yang ada di AWS wilayah yang sama dengan broker
   - Konfirmasikan sintaks ARN benar
   - Pastikan peran IAM memiliki izin yang sesuai untuk mengakses sumber daya

4. Validasi perbaikan menggunakan titik akhir API validasi akses ARN sebelum memperbarui konfigurasi broker.
5. Perbarui konfigurasi broker dan reboot broker.

Sumber daya terkait

Sumber daya Amazon MQ

Tabel berikut mencantumkan sumber daya yang bermanfaat untuk bekerja dengan Amazon MQ.

Sumber Daya	Deskripsi
Referensi API Amazon MQ REST	Deskripsi sumber daya REST, permintaan contoh, metode HTTP, skema, parameter, dan kesalahan yang dikembalikan layanan.
Amazon MQ dalam Referensi Perintah AWS CLI	Deskripsi AWS CLI perintah yang dapat Anda gunakan untuk bekerja dengan broker pesan.
Amazon MQ di Panduan Pengguna AWS CloudFormation	Sumber daya `AWS::Amazon MQ::Broker` memungkinkan Anda membuat broker Amazon MQ, menambahkan perubahan konfigurasi atau memodifikasi pengguna untuk broker tertentu, mengembalikan informasi tentang broker tertentu, dan menghapus broker tertentu. Sumber daya `AWS::Amazon MQ::Configuration` memungkinkan Anda membuat konfigurasi Amazon MQ, menambahkan perubahan konfigurasi atau memodifikasi pengguna, dan mengembalikan informasi tentang konfigurasi tertentu.
Wilayah dan Titik Akhir	Informasi tentang wilayah dan titik akhir Amazon MQ
Halaman Produk	Halaman web utama untuk informasi tentang Amazon MQ.

Sumber Daya	Deskripsi
Forum Diskusi	Forum berbasis komunitas untuk developer guna membahas pertanyaan teknis terkait Amazon MQ.
AWS Informasi Support Premium	Halaman web utama untuk informasi tentang AWS Premium Support, saluran dukungan respons cepat untuk membantu Anda membangun dan menjalankan aplikasi pada layanan infrastruktur one-on-one AWS

Sumber daya Amazon MQ for ActiveMQ

Tabel berikut mencantumkan sumber daya yang bermanfaat untuk bekerja dengan Apache ActiveMQ.

Sumber Daya	Deskripsi
Panduan Memulai Apache ActiveMQ	Dokumentasi resmi dari Apache ActiveMQ.
ActiveMQ dalam Aksi	Panduan untuk Apache ActiveMQ yang mencakup anatomi pesan, konektor, persistensi pesan, autentikasi, dan otorisasi JMS.
Klien Lintas Bahasa	Daftar bahasa pemrograman dan pustaka Apache ActiveMQ yang sesuai. Lihat juga Klien ActiveMQ dan Klien QpidJMS.

Sumber daya Amazon MQ for RabbitMQ

Tabel berikut mencantumkan sumber daya yang bermanfaat untuk bekerja dengan RabbitMQ.

Sumber Daya	Deskripsi
Panduan Memulai RabbitMQ	Dokumentasi resmi RabbitMQ.

Sumber Daya	Deskripsi
Perpustakaan Klien RabbitMQ dan Alat Pengembang	Panduan untuk pustaka klien yang didukung secara resmi dan alat developer untuk bekerja dengan RabbitMQ menggunakan berbagai bahasa pemrograman dan platform.
Praktik Terbaik RabbitMQ	Panduan CloudAMQP tentang praktik terbaik dan rekomendasi untuk bekerja dengan RabbitMQ.

Catatan rilis Amazon MQ

Tabel berikut mencantumkan rilis fitur dan peningkatan Amazon MQ.

Date	Pembaruan Dokumentasi
Februari 19, 2026	Amazon MQ Amazon MQ sekarang mendukung ActiveMQ 5.19, rilis versi mesin minor baru. Untuk informasi selengkapnya, silakan lihat • Halaman Rilis ActiveMQ 5.19 • Mengelola versi mesin Amazon MQ for ActiveMQ • Meningkatkan versi mesin broker Amazon MQ • Menggunakan file konfigurasi Spring XML
Januari 22, 2026	Amazon MQ sekarang mendukung plugin pertukaran topik JMS untuk broker di RabbitMQ 4.2 dan di atasnya. Anda dapat menggunakan klien JMS RabbitMQ resmi untuk menjalankan beban kerja JMS di Amazon MQ untuk broker RabbitMQ. Ini mendukung JMS 1.1, 2.0 dan 3.1. Untuk informasi selengkapnya, silakan lihat • Spesifikasi JMS 2.0 resmi (kompatibel dengan dan diperpanjang JMS 1.1) • Spesifikasi resmi JMS 3.1 • Batasan klien RabbitMQ JMS • Menghubungkan aplikasi JMS Anda ke Amazon MQ untuk broker RabbitMQ
Januari 8, 2026	Amazon MQ sekarang mendukung otentikasi sertifikat SSL untuk broker di RabbitMQ 4.2 dan di atasnya menggunakan sertifikat klien X.509, dan konfigurasi TLS (mTLS) bersama. Anda dapat mengonfigurasi otentikasi sertifikat SSL dan mTL melalui Konsol Manajemen AWS,, AWS CloudForm ation AWS CLI, atau AWS CDK di semua tempat Amazon Wilayah AWS MQ tersedia.

Date	Pembaruan Dokumentasi
	Untuk informasi selengkapnya, lihat otentikasi sertifikat SSL dan Mengonfigurasi mTL.
Januari 6, 2026	Amazon MQ sekarang mendukung otentikasi HTTP dan otorisasi untuk broker di RabbitMQ 4.2 dan di atasnya dengan server HTTP eksternal. Anda dapat mengonfigurasi otentikasi HTTP melalui Konsol Manajemen AWS AWS CloudFormation,, AWS CLI, atau AWS CDK di semua Wilayah AWS tempat Amazon MQ tersedia. Untuk informasi selengkapnya, lihat otentikasi dan otorisasi HTTP.
November 20, 2025	Amazon MQ sekarang mendukung RabbitMQ 4.2, rilis versi utama baru yang memperkenalkan dukungan asli untuk protokol AMQP 1.0, toko metadata berbasis Raft baru Khepri, sekop lokal, dan prioritas pesan untuk antrian kuorum. RabbitMQ 4.2 juga mencakup berbagai perbaikan bug dan peningkatan kinerja untuk throughput dan manajemen memori. Meskipun versi ini memperkenalkan fitur-fitur baru, ada beberapa perubahan yang melanggar. Untuk informasi selengkapnya, silakan lihat • KelinciMQ 4 • Catatan rilis RabbitMQ sumber terbuka • Mengkonfigurasi batas sumber daya • Protokol yang Didukung • Peningkatan Versi Amazon MQ
November 18, 2024	Amazon MQ sekarang mendukung instans m7g bertenaga Graviton3 untuk RabbitMQ dalam berbagai ukuran dari sedang hingga 16xlarge di Afrika (Cape Town). Untuk informasi selengkapnya, lihat Amazon MQ untuk jenis instans broker RabbitMQ.

Date	Pembaruan Dokumentasi
November 17, 2025	Amazon MQ sekarang mendukung otentikasi dan otorisasi LDAP untuk broker RabbitMQ dengan layanan direktori LDAP eksternal. Anda dapat mengonfig urasi LDAP melalui Konsol Manajemen AWS AWS CloudFormation,, AWS CLI, atau AWS CDK di semua Wilayah AWS tempat Amazon MQ tersedia. Untuk informasi selengkapnya, lihat Otentikasi dan otorisasi LDAP untuk Amazon MQ untuk RabbitMQ.
Oktober 22, 2025	Amazon MQ sekarang tersedia di Wilayah Asia Pasifik (Selandia Baru). Untuk informasi tentang wilayah yang tersedia, lihat AWS Wilayah dan Titik Akhir dalam panduan Referensi AWS Umum.
September 3, 2025	Amazon MQ sekarang mendukung otentikasi OAuth 2.0 dan otorisasi untuk broker RabbitMQ dengan penyedia identitas publik (). IdPs Anda dapat mengonfigurasi OAuth 2.0 melalui Konsol Manajemen AWS AWS CloudForm ation, AWS CLI,, atau AWS CDK di semua Wilayah AWS tempat Amazon MQ tersedia. Untuk informasi selengkapnya, lihat OAuth 2.0 otentikasi dan otorisasi untuk Amazon MQ untuk RabbitMQ.
Juli 22, 2025	Amazon MQ sekarang mendukung `m7g` instans bertenaga Graviton3 untuk RabbitMQ dalam berbagai ukuran dari sedang hingga 16xlarge. Cluster RabbitMQ yang berjalan pada `m7g` instans memberikan kapasitas beban kerja hingga 50% lebih tinggi dan peningkatan throughput hingga 85% dibanding kan Amazon MQ yang sebanding untuk cluster RabbitMQ yang berjalan pada instans. `m5` `M7g`instance juga memiliki ukuran volume disk yang dioptimalkan yang bervariasi menurut ukuran instans. Untuk informasi selengkapnya, lihat Broker instance types. `M7g`Instans di Amazon MQ tersedia saat ini di semua Wilayah yang tersedia secara umum kecuali Afrika (Cape Town), Kanada Barat (Calgary), dan Wilayah Eropa (Milan).

Date	Pembaruan Dokumentasi
Juli 8, 2025	Amazon MQ sekarang tersedia di Wilayah Asia Pasifik (Taipei). Untuk informasi tentang wilayah yang tersedia, lihat AWS Wilayah dan Titik Akhir dalam panduan Referensi AWS Umum.
April 22, 2025	Anda sekarang dapat menghapus konfigurasi broker Amazon MQ menggunak an API. `DeleteConfiguration` Untuk informasi selengkapnya, lihat Konfigurasi di Referensi API Amazon MQ.
April 16, 2025	Amazon MQ untuk RabbitMQ sekarang mendukung penggunaan titik akhir dual-stack (IPv4 dan IPv6) untuk terhubung ke broker publik dan swasta. Untuk informasi selengkapnya, lihat Connecting to Amazon MQ dan Configuri ng a private Amazon MQ broker.
April 7, 2025	Amazon MQ sekarang tersedia di Wilayah Asia Pasifik (Thailand) dan Meksiko (Tengah). Untuk informasi tentang wilayah yang tersedia, lihat AWS Wilayah dan Titik Akhir dalam panduan Referensi AWS Umum.
Februari 13, 2025	Titik akhir Amazon MQ API FIPS sekarang tersedia di Wilayah Kanada (Tengah) dan Kanada Barat (Calgary). Untuk informasi selengkapnya tentang penggunaan titik akhir FIPS dengan Amazon MQ API, lihat. Connecting to Amazon MQ Untuk informasi tentang wilayah yang tersedia, lihat AWS Wilayah dan Titik Akhir dalam panduan Referensi AWS Umum.

Date	Pembaruan Dokumentasi
Februari 12, 2025	Amazon MQ mengumumkan tanggal akhir dukungan jenis instans berikut: Broker instance types • `mq.t2.micro` ActiveMQ: 12 Mei 2025 • `mq.m4.large` ActiveMQ: 12 Mei 2025 Anda tidak dapat membuat broker pada `mq.t2.micro` atau `mq.m4.large` setelah 17 Maret 2025.
Desember 10, 2024	Amazon MQ sekarang mendukung penggunaan AWS PrivateLink untuk menghubungkan antara awan pribadi virtual Anda (VPCs) dan Amazon MQ API tanpa mengekspos lalu lintas Anda ke internet publik. Untuk informasi selengkapnya, lihat the section called “Connect ke Amazon MQ menggunakan AWS PrivateLink”.
November 18, 2024	Amazon MQ sekarang tersedia di Wilayah Asia Pasifik (Malaysia). Untuk informasi tentang wilayah yang tersedia, lihat AWS Wilayah dan Titik Akhir dalam panduan Referensi AWS Umum.
November 14, 2024	Amazon MQ mengumumkan tanggal dukungan akhir versi mesin berikut: Mengelola versi mesin Amazon MQ for ActiveMQ • ActiveMQ 5.17:16 Juni 2025 Mengelola Amazon MQ untuk versi mesin RabbitMQ • RabbitMQ 3.11:17 Februari 2025 • RabbitMQ 3.12:17 Maret 2025 Untuk informasi selengkapnya tentang memutakhirkan ke versi terbaru, lihat Meningkatkan versi mesin broker Amazon MQ

Date	Pembaruan Dokumentasi
November 13, 2024	Amazon MQ sekarang mendukung titik akhir layanan dual-stack yang dapat Anda sambungkan menggunakan salah satu atau. IPv4 IPv6 Amazon MQ dual-stack Regional endpoint layanan dapat diselesaikan dengan keduanya dan catatan DNS. A AAAA Untuk informasi selengkapnya, lihat ???.
Juli 25, 2024	Amazon MQ sekarang mendukung ActiveMQ 5.18, rilis versi mesin minor baru. Untuk informasi selengkapnya, lihat berikut ini: • ActiveMQ 5.18 Halaman Rilis • Mengelola versi mesin Amazon MQ for ActiveMQ • Meningkatkan versi mesin broker Amazon MQ • Menggunakan file konfigurasi Spring XML
Juli 22, 2024	Amazon MQ sekarang mendukung antrian kuorum hanya pada broker menggunakan versi 3.13 ke atas. Quorum queues adalah tipe antrian FIFO yang direplikasi yang menggunakan algoritma konsensus Raft untuk menjaga konsistensi data. Antrian kuorum menyediakan penanganan pesan racun, yang dapat membantu Anda mengelola pesan yang belum diproses. Untuk memulai dengan antrian kuorum, lihat. Antrian kuorum untuk RabbitMQ di Amazon MQ

Date	Pembaruan Dokumentasi
Juli 2, 2024	Amazon MQ untuk RabbitMQ sekarang mendukung RabbitMQ 3.13, rilis versi minor. Untuk semua broker yang menggunakan engine versi 3.13 ke atas, Amazon MQ mengelola peningkatan ke versi patch terbaru yang didukung selama jendela pemeliharaan. Untuk informasi selengkapnya, lihat Meningkatkan versi mesin broker Amazon MQ. Amazon MQ untuk pedoman ukuran RabbitMQtelah diperbarui untuk memasukkan batas baru untuk antrian, konsumen per saluran, dan sekop untuk broker menggunakan mesin versi 3.13. Untuk informasi selengkapnya tentang perbaikan dan fitur dalam rilis ini, lihat catatan rilis RabbitMQ 3.13 di repositori server RabbitMQ. GitHub Untuk informasi selengkapnya tentang Amazon MQ yang didukung untuk versi RabbitMQ dan peningkatan broker, lihat. Mengelola Amazon MQ untuk versi mesin RabbitMQ
Juni 10, 2024	Amazon MQ sekarang tersedia di Wilayah Kanada Barat (Calgary). Untuk informasi tentang wilayah yang tersedia, lihat AWS Wilayah dan Titik Akhir dalam panduan Referensi AWS Umum.
10 Mei 2024	Kalender dukungan versi Amazon MQ menunjukkan kapan versi mesin broker mencapai akhir dukungan. Ketika versi mesin mencapai akhir dukungan, Amazon MQ memperbarui semua broker pada versi ke versi minor yang didukung berikutnya secara otomatis. Amazon MQ menyediakan setidaknya pemberitahuan 90 hari sebelum versi mesin mencapai akhir dukungan. Untuk melihat kalender dukungan versi dan akhir dukungan, lihat berikut ini: • Mengelola versi mesin Amazon MQ for ActiveMQ • Mengelola Amazon MQ untuk versi mesin RabbitMQ Anda juga dapat mengaktifkan upgrade versi minor otomatis untuk broker Anda untuk memperbarui ke versi patch berikutnya selama jendela pemeliharaan. Untuk informasi selengkapnya, lihat Meningkatkan versi mesin broker Amazon MQ

Date	Pembaruan Dokumentasi
9 Mei 2024	Amazon MQ untuk RabbitMQ sekarang mendukung RabbitMQ 3.12, rilis versi minor. Semua broker di 3.12.13 dan di atasnya menggunakan Antrian Klasik versi 2 (CQv2), dan semua antrian pada 3.12.13 ke atas berperilaku sebagai antrian malas. Kami merekomendasikan broker pada versi sebelum 3.12.13 mengaktifkan CQv2 dan antrian malas, atau meningkatkan ke versi terbaru Amazon MQ untuk RabbitMQ. Untuk informasi selengkapnya tentang perbaikan dan fitur dalam rilis ini, lihat berikut ini: • Catatan rilis RabbitMQ 3.12 pada repositori server RabbitMQ. GitHub Untuk informasi selengkapnya tentang Amazon MQ yang didukung untuk versi RabbitMQ dan peningkatan broker, lihat. Mengelola Amazon MQ untuk versi mesin RabbitMQ
Maret 4, 2024	Amazon MQ untuk RabbitMQ sekarang mendukung RabbitMQ 3.11.28. Untuk informasi selengkapnya tentang perbaikan dan fitur dalam rilis ini, lihat berikut ini: • Catatan rilis RabbitMQ 3.11.28 pada repositori server RabbitMQ GitHub • RabbitMQ changelog Untuk informasi selengkapnya tentang Amazon MQ yang didukung untuk versi RabbitMQ dan peningkatan broker, lihat. Mengelola Amazon MQ untuk versi mesin RabbitMQ
Januari 19, 2024	Amazon MQ untuk RabbitMQ tidak mendukung nama pengguna “tamu”, dan akan menghapus akun tamu default saat Anda membuat broker baru. Amazon MQ juga akan secara berkala menghapus akun yang dibuat pelanggan yang disebut “tamu”.

Date	Pembaruan Dokumentasi
15 Desember 2023	Amazon MQ sekarang tersedia di Wilayah Israel (Tel Aviv). Untuk informasi tentang wilayah yang tersedia, lihat AWS Wilayah dan Titik Akhir dalam panduan Referensi AWS Umum.
Desember 11, 2023	Amazon MQ untuk RabbitMQ sekarang mendukung RabbitMQ 3.10.25. Untuk informasi selengkapnya tentang perbaikan dan fitur dalam rilis ini, lihat berikut ini: • Catatan rilis RabbitMQ 3.10.25 pada repositori server RabbitMQ GitHub • RabbitMQ changelog Untuk informasi selengkapnya tentang Amazon MQ yang didukung untuk versi RabbitMQ dan peningkatan broker, lihat. Mengelola Amazon MQ untuk versi mesin RabbitMQ
26 Oktober 2023	Amazon MQ telah merilis ActiveMQ minor versi terbaru 5.15.16, 5.16.7, 5.17.6 dengan pembaruan kritis. Kami telah menghentikan versi minor ActiveMQ yang lebih lama dan akan memperbarui semua broker pada versi 5.15 ke 5.15.16, atau 5.16 ke 5.16.7 dan 5.17 ke 5.17.6. Untuk informasi lebih lanjut tentang memperbarui broker ActiveMQ Anda, lihat. Mengelola versi mesin Amazon MQ for ActiveMQ
27 September 2023	Amazon MQ untuk RabbitMQ sekarang mendukung RabbitMQ 3.11.20. Untuk informasi selengkapnya tentang perbaikan dan fitur dalam rilis ini, lihat berikut ini: • Catatan rilis RabbitMQ 3.11.20 pada repositori server RabbitMQ GitHub • RabbitMQ changelog Untuk informasi selengkapnya tentang Amazon MQ yang didukung untuk versi RabbitMQ dan peningkatan broker, lihat. Mengelola Amazon MQ untuk versi mesin RabbitMQ

Date	Pembaruan Dokumentasi
Juli 27, 2023	Amazon MQ untuk RabbitMQ sekarang mendukung RabbitMQ 3.11.16 Untuk informasi selengkapnya tentang perbaikan dan fitur dalam rilis ini, lihat berikut ini: • Catatan rilis RabbitMQ 3.11.16 pada repositori server RabbitMQ GitHub • RabbitMQ changelog Untuk informasi selengkapnya tentang Amazon MQ yang didukung untuk versi RabbitMQ dan peningkatan broker, lihat. Mengelola Amazon MQ untuk versi mesin RabbitMQ
Juli 27, 2023	Amazon MQ untuk RabbitMQ sekarang mendukung pembuatan dan penerapan konfigurasi ke broker RabbitMQ Anda. Untuk informasi lebih lanjut tentang menambahkan konfgurasi ke broker Anda, lihat. RabbitMQ Broker Configurations Untuk informasi selengkapnya tentang fitur ini, lihat: • Kebijakan operator • Perubahan kebijakan operator
23 Juni 2023	Amazon MQ sekarang mendukung ActiveMQ 5.17.3, rilis versi mesin minor baru. Rilis ini mendukung fitur replikasi data Cross-region (CRDR) baru dari Amazon MQ. Untuk informasi selengkapnya, lihat berikut ini: • Untuk memulai CRDR, lihat Replikasi data lintas wilayah untuk Amazon MQ untuk ActiveMQ di Panduan Pengembang. • ActiveMQ 5.17.3 Halaman Rilis • Mengelola versi mesin Amazon MQ for ActiveMQ • Meningkatkan versi mesin broker Amazon MQ • Menggunakan file konfigurasi Spring XML

Date	Pembaruan Dokumentasi
Juni 21, 2023	Amazon MQ untuk ActiveMQ sekarang menawarkan fitur replikasi data lintas wilayah (CRDR) yang memungkinkan replikasi pesan asinkron dari broker utama di Wilayah utama ke broker replika di Wilayah replika. AWS Jika broker utama di Wilayah primer gagal, Anda dapat mempromosikan broker replika di Wilayah sekunder ke primer dengan memulai peralihan atau failover. Untuk memulai CRDR, lihat Replikasi data lintas wilayah untuk Amazon MQ untuk ActiveMQ di Panduan Pengembang.
18 Mei 2023	Amazon MQ sekarang tersedia di wilayah berikut: • Asia Pacific (Melbourne) • Asia Pasifik (Hyderabad) • Eropa (Spanyol) • Europe (Zurich) Untuk informasi tentang wilayah yang tersedia, lihat AWS Wilayah dan Titik Akhir dalam panduan Referensi AWS Umum.
April 14, 2023	Amazon MQ untuk RabbitMQ sekarang mendukung RabbitMQ versi 3.9.27. Untuk informasi selengkapnya tentang perbaikan dan fitur dalam rilis ini, lihat berikut ini: • Catatan rilis RabbitMQ 3.9.27 pada repositori server RabbitMQ GitHub • RabbitMQ changelog Untuk informasi selengkapnya tentang Amazon MQ yang didukung untuk versi RabbitMQ dan peningkatan broker, lihat. Mengelola Amazon MQ untuk versi mesin RabbitMQ

Date	Pembaruan Dokumentasi
April 14, 2023	Amazon MQ untuk RabbitMQ sekarang mendukung RabbitMQ versi 3.10.20. Untuk informasi selengkapnya tentang perbaikan dan fitur dalam rilis ini, lihat berikut ini: • Catatan rilis RabbitMQ 3.10.20 pada repositori server RabbitMQ GitHub • RabbitMQ changelog Untuk informasi selengkapnya tentang Amazon MQ yang didukung untuk versi RabbitMQ dan peningkatan broker, lihat. Mengelola Amazon MQ untuk versi mesin RabbitMQ
31 Maret 2023	Amazon MQ untuk RabbitMQ telah menonaktifkan mesin RabbitMQ versi 3.10.17 Amazon MQ untuk tim RabbitMQ, dan pengelola open source RabbitMQ, telah mengidentifikasi masalah dengan konsol manajemen RabbitMQ pada versi 3.10.17. Amazon MQ telah mencabut versi ini. Untuk mengurangi dampak dari masalah ini, buat broker baru dengan versi 3.10.10 sementara kami bekerja untuk mendukung versi patch baru RabbitMQ. Sebaiknya aktifkan opsi pemutakhiran Versi untuk secara otomatis mendapatkan perbaikan bug terbaru, pembaruan keamanan, dan peningkatan kinerja. Untuk informasi selengkapnya tentang Amazon MQ yang tersedia untuk versi RabbitMQ, lihat Amazon MQ untuk versi mesin RabbitMQ.

Date	Pembaruan Dokumentasi
1 Maret 2023	Amazon MQ untuk RabbitMQ sekarang mendukung RabbitMQ versi 3.10.17. Untuk informasi selengkapnya tentang perbaikan dan fitur dalam rilis ini, lihat berikut ini: • Catatan rilis RabbitMQ 3.10.17 pada repositori server RabbitMQ GitHub • RabbitMQ changelog Untuk informasi selengkapnya tentang Amazon MQ yang didukung untuk versi RabbitMQ dan peningkatan broker, lihat. Mengelola Amazon MQ untuk versi mesin RabbitMQ
21 Februari 2023	Amazon MQ untuk RabbitMQ sekarang terintegrasi dengan AWS Key Management Service (KMS) untuk menawarkan enkripsi sisi server. Sekarang Anda dapat memilih CMK yang dikelola pelanggan Anda sendiri, atau menggunakan kunci KMS AWS terkelola di akun Anda AWS KMS . Untuk informasi selengkapnya, lihat Enkripsi saat diam. Amazon MQ mendukung penggunaan AWS KMS kunci dengan cara berikut. • Kunci KMS milik Amazon MQ (default) - Kunci dimiliki dan dikelola oleh Amazon MQ dan tidak ada di akun Anda. • AWS kunci KMS AWS terkelola - Kunci KMS terkelola (`aws/mq`) adalah kunci KMS di akun Anda yang dibuat, dikelola, dan digunakan atas nama Anda oleh Amazon MQ. • Pilih kunci KMS yang ada dan dikelola pelanggan — Kunci KMS yang dikelola pelanggan dibuat dan dikelola oleh Anda di AWS Key Management Service (KMS).

Date	Pembaruan Dokumentasi
13 Januari 2023	Amazon MQ untuk RabbitMQ sekarang mendukung RabbitMQ versi 3.8.34. Untuk informasi selengkapnya tentang perbaikan dan fitur dalam rilis ini, lihat berikut ini: • Catatan rilis RabbitMQ 3.8.34 pada repositori server RabbitMQ GitHub • RabbitMQ changelog Untuk informasi selengkapnya tentang Amazon MQ yang didukung untuk versi RabbitMQ dan peningkatan broker, lihat. Mengelola Amazon MQ untuk versi mesin RabbitMQ
15 Desember 2022	Amazon MQ untuk RabbitMQ sekarang mendukung RabbitMQ versi 3.9.24. Untuk informasi selengkapnya tentang perbaikan dan fitur dalam rilis ini, lihat berikut ini: • Catatan rilis RabbitMQ 3.9.24 pada repositori server RabbitMQ GitHub • RabbitMQ changelog Untuk informasi selengkapnya tentang Amazon MQ yang didukung untuk versi RabbitMQ dan peningkatan broker, lihat. Mengelola Amazon MQ untuk versi mesin RabbitMQ
13 Desember 2022	Amazon MQ sekarang tersedia di Wilayah Timur Tengah (UEA). Untuk informasi tentang wilayah yang tersedia, lihat AWS Wilayah dan Titik Akhir dalam panduan Referensi AWS Umum.

Date	Pembaruan Dokumentasi
November 14, 2022	Amazon MQ untuk RabbitMQ sekarang mendukung 3.10, rilis versi mesin utama. Anda sekarang dapat mengaktifkan Antrian klasik versi 2 (CQv2) pada antrian RabbitMQ Anda. Pembaruan langsung dari 3.8 ke 3.10 tidak didukung. Untuk informasi selengkapnya, lihat berikut ini: • Catatan rilis RabbitMQ 3.10.10 • RabbitMQ changelog Untuk informasi selengkapnya tentang Amazon MQ yang didukung untuk versi RabbitMQ dan peningkatan broker, lihat. Mengelola Amazon MQ untuk versi mesin RabbitMQ
9 November 2022	Amazon MQ sekarang mendukung ActiveMQ 5.17.2, rilis versi mesin minor baru. Untuk informasi selengkapnya, lihat berikut ini: • ActiveMQ 5.17.2 Halaman Rilis • Mengelola versi mesin Amazon MQ for ActiveMQ • Meningkatkan versi mesin broker Amazon MQ • Menggunakan file konfigurasi Spring XML
17 Agustus 2022	Amazon MQ sekarang mendukung ActiveMQ 5.17.1, rilis versi mesin utama baru. Untuk informasi selengkapnya, lihat berikut ini: • ActiveMQ 5.17.1 Halaman Rilis • Mengelola versi mesin Amazon MQ for ActiveMQ • Meningkatkan versi mesin broker Amazon MQ • Menggunakan file konfigurasi Spring XML

Date	Pembaruan Dokumentasi
14 Juli 2022	Amazon MQ sekarang mendukung ActiveMQ 5.16.5, rilis versi mesin minor. Untuk informasi selengkapnya, lihat berikut ini: • ActiveMQ 5.16.5 Halaman Rilis • Mengelola versi mesin Amazon MQ for ActiveMQ • Menggunakan file konfigurasi Spring XML • Meningkatkan versi mesin broker Amazon MQ
4 Mei, 2022	Amazon MQ menambahkan bahasa inklusif untuk `networkConnector` elemen dalam konfigurasi broker. • Membuat dan mengonfigurasi jaringan broker Amazon MQ
April 25, 2022	Amazon MQ Rilis ini menambahkan status `CRITICAL_ACTION_REQUIRED` broker dan properti `ActionRequired` API. `CRITICAL_ACTION_REQUIRED` memberi tahu Anda ketika broker Anda terdegradasi. `ActionRequired` memberi Anda kode yang dapat Anda gunakan untuk menemukan petunjuk di Panduan Pengembang tentang cara mengatasi masalah. • Pemecahan masalah • `ActionRequired` dokumentasi di Referensi API Amazon MQ.
20 April 2022	Amazon MQ sekarang mendukung ActiveMQ 5.16.4, rilis versi mesin minor. Untuk informasi selengkapnya, lihat berikut ini: • ActiveMQ 5.16.4 Halaman Rilis • Mengelola versi mesin Amazon MQ for ActiveMQ • Menggunakan file konfigurasi Spring XML • Meningkatkan versi mesin broker Amazon MQ
1 Maret 2022	Amazon MQ sekarang tersedia di Wilayah Asia Pasifik (Jakarta). Untuk informasi tentang wilayah yang tersedia, lihat AWS Wilayah dan Titik Akhir dalam panduan Referensi AWS Umum.

Date	Pembaruan Dokumentasi
25 Februari 2022	Amazon MQ untuk RabbitMQ sekarang mendukung RabbitMQ versi 3.8.27. Untuk informasi selengkapnya tentang perbaikan dan fitur dalam rilis ini, lihat berikut ini: • Catatan rilis RabbitMQ 3.8.27 pada repositori server RabbitMQ GitHub • RabbitMQ changelog Untuk informasi selengkapnya tentang Amazon MQ yang didukung untuk versi RabbitMQ dan peningkatan broker, lihat. Mengelola Amazon MQ untuk versi mesin RabbitMQ
16 Februari 2022	Amazon MQ sekarang tersedia di Wilayah Afrika (Cape Town). Untuk informasi tentang wilayah yang tersedia, lihat AWS Wilayah dan Titik Akhir dalam panduan Referensi AWS Umum.

Date	Pembaruan Dokumentasi
14 Februari 2022	Amazon MQ untuk RabbitMQ sekarang mendukung RabbitMQ versi 3.9.13. Upgrade versi minor otomatis tidak dapat digunakan untuk meningkatkan dari Rabbit 3.8 ke 3.9. Untuk melakukannya, tingkatkan broker Anda secara manual. Untuk informasi lebih lanjut tentang fitur baru yang diperkenalkan di RabbitMQ 3.9, lihat halaman catatan rilis untuk versi 3.9.0 di situs web. GitHub **Note** Saat ini, Amazon MQ tidak mendukung aliran, atau menggunakan logging terstruktur di JSON, diperkenalkan di RabbitMQ 3.9. Untuk informasi selengkapnya tentang perbaikan dan fitur dalam rilis ini, lihat berikut ini: • Catatan rilis RabbitMQ 3.9.13 pada repositori server RabbitMQ GitHub • RabbitMQ changelog Untuk informasi selengkapnya tentang Amazon MQ yang didukung untuk versi RabbitMQ dan peningkatan broker, lihat. Mengelola Amazon MQ untuk versi mesin RabbitMQ
Februari 07, 2022	Amazon MQ untuk RabbitMQ memperkenalkan metrik broker baru, memungkinkan Anda memantau pemanfaatan sumber daya rata-rata di ketiga node dalam penerapan cluster. Untuk informasi selengkapnya, lihat berikut ini: • the section called “Metrik untuk RabbitMQ”

Date	Pembaruan Dokumentasi
18 Januari 2022	Amazon MQ untuk RabbitMQ sekarang mendukung RabbitMQ versi 3.8.26. Untuk informasi selengkapnya tentang perbaikan dan fitur dalam rilis ini, lihat berikut ini: • Catatan rilis RabbitMQ 3.8.26 pada repositori server RabbitMQ GitHub • RabbitMQ changelog Untuk informasi selengkapnya tentang Amazon MQ yang didukung untuk versi RabbitMQ dan peningkatan broker, lihat. Mengelola Amazon MQ untuk versi mesin RabbitMQ
Januari 13, 2022	Amazon MQ memperkenalkan kode `RABBITMQ_MEMORY_ALARM` status untuk memberi tahu Anda ketika broker Anda telah menaikkan alarm memori tinggi dan dalam keadaan tidak sehat. Amazon MQ memberikan informasi dan rekomendasi terperinci untuk membantu Anda mendiagnosis, menyelesa ikan, dan mencegah alarm memori tinggi. Untuk informasi selengkapnya, lihat hal berikut. • the section called “ `RABBITMQ_MEMORY_ALARM` ”
6 Januari 2022	Saat Anda mengonfigurasi CloudWatch Log untuk Amazon MQ untuk broker ActiveMQ, Amazon MQ mendukung penggunaan `aws:SourceAccount` dan kunci konteks kondisi global dalam kebijakan berbasis `aws:SourceArn` sumber daya IAM untuk mencegah masalah deputi yang membingungkan. Untuk informasi selengkapnya, lihat hal berikut. • the section called “Pencegahan "confused deputy" lintas layanan”

Date	Pembaruan Dokumentasi
Desember 20, 2021	Amazon MQ untuk ActiveMQ memperkenalkan satu set metrik baru, memungkinkan Anda untuk memantau jumlah maksimum koneksi yang dapat Anda buat ke broker Anda menggunakan berbagai protokol transport asi yang didukung, serta metrik baru tambahan yang memungkinkan Anda memantau jumlah node yang terhubung ke broker Anda di jaringan broker. Untuk informasi selengkapnya, lihat hal berikut. • the section called “Metrik untuk ActiveMQ”
November 16, 2021	Amazon MQ untuk RabbitMQ sekarang mendukung RabbitMQ versi 3.8.23. Untuk informasi selengkapnya tentang perbaikan dan fitur dalam rilis ini, lihat berikut ini: • Catatan rilis RabbitMQ 3.8.23 pada repositori server RabbitMQ GitHub • RabbitMQ changelog Untuk informasi selengkapnya tentang Amazon MQ yang didukung untuk versi RabbitMQ dan peningkatan broker, lihat. Mengelola Amazon MQ untuk versi mesin RabbitMQ
Oktober 12, 2021	Amazon MQ sekarang mendukung ActiveMQ 5.16.3, rilis versi mesin minor. Untuk informasi selengkapnya, lihat berikut ini: • ActiveMQ 5.16.3 Halaman Rilis • Mengelola versi mesin Amazon MQ for ActiveMQ • Meningkatkan versi mesin broker Amazon MQ • Menggunakan file konfigurasi Spring XML

Date	Pembaruan Dokumentasi
8 September 2021	Amazon MQ untuk RabbitMQ sekarang mendukung RabbitMQ versi 3.8.22. Rilis ini mencakup perbaikan untuk masalah dengan antrian menggunakan TTL per pesan (waktu untuk hidup), diidentifikasi dalam versi yang didukung sebelumnya, RabbitMQ 3.8.17. Kami merekomendasikan untuk meningkatkan broker Anda yang ada ke versi 3.8.22. Untuk informasi selengkapnya tentang perbaikan dan fitur dalam rilis ini, lihat berikut ini: • Catatan rilis RabbitMQ 3.8.22 pada repositori server RabbitMQ GitHub • RabbitMQ changelog Untuk informasi selengkapnya tentang Amazon MQ yang didukung untuk versi RabbitMQ dan peningkatan broker, lihat Mengelola Amazon MQ untuk versi mesin RabbitMQ
25 Agustus 2021	Amazon MQ untuk RabbitMQ telah menonaktifkan sementara mesin RabbitMQ versi 3.8.17 karena masalah yang diidentifikasi dengan antrian menggunakan per-pesan (TTL). time-to-live Kami merekomendasikan menggunakan versi 3.8.11.
29 Juli 2021	Amazon MQ untuk RabbitMQ sekarang mendukung RabbitMQ versi 3.8.17. Untuk informasi selengkapnya tentang perbaikan dan fitur yang ada dalam pembaruan ini, lihat: • Catatan rilis RabbitMQ 3.8.17 pada repositori server RabbitMQ GitHub • RabbitMQ changelog • Mengelola Amazon MQ untuk versi mesin RabbitMQ
16 Juli 2021	Anda sekarang dapat menyesuaikan jendela pemeliharaan broker Amazon MQ menggunakan Konsol Manajemen AWS, AWS CLI, atau Amazon MQ API. Untuk mempelajari lebih lanjut tentang jendela pemeliharaan broker, lihat yang berikut ini. • Menjadwalkan jendela pemeliharaan untuk broker Amazon MQ

Date	Pembaruan Dokumentasi
6 Juli 2021	Amazon MQ untuk RabbitMQ memperkenalkan dukungan untuk jenis pertukaran Hash yang Konsisten. Pertukaran Hash yang konsisten merutekan pesan ke antrian berdasarkan nilai hash yang dihitung dari kunci perutean pesan. Untuk informasi selengkapnya, lihat berikut ini: • Plugin pertukaran Hash yang konsisten • Jenis Pertukaran Hash Konsisten RabbitMQ pada repositori RabbitMQ GitHub
7 Juni 2021	Amazon MQ kini mendukung ActiveMQ 5.16.2, rilis baru versi mesin utama. Untuk informasi lebih lanjut, lihat hal berikut: • ActiveMQ 5.16.2 Halaman Rilis • Mengelola versi mesin Amazon MQ for ActiveMQ • Meningkatkan versi mesin broker Amazon MQ • Menggunakan file konfigurasi Spring XML
26 Mei 2021	Amazon MQ untuk RabbitMQ kini tersedia di Wilayah China (Beijing) dan China (Ningxia). Untuk informasi tentang wilayah yang tersedia, lihat Wilayah dan Titik Akhir AWS.
18 Mei 2021	Amazon MQ for RabbitMQ mengimplementasikan broker default. Ketika Anda pertama kali membuat broker, Amazon MQ membuat serangkai an kebijakan broker dan batas vhost berdasarkan tipe instans dan mode deployment yang Anda pilih, untuk mengoptimalkan performa broker. Untuk informasi lebih lanjut, lihat yang berikut ini: https://docs.aws.amazon.com//amazon-mq/latest/developer-guide/rabbitmq-defaults.html
5 Mei 2021	Amazon MQ kini mendukung ActiveMQ 5.15.15. Untuk informasi lebih lanjut, lihat hal berikut: • ActiveMQ 5.15.15 Halaman Rilis • Mengelola versi mesin Amazon MQ for ActiveMQ • Menggunakan file konfigurasi Spring XML

Date	Pembaruan Dokumentasi
5 Mei 2021	Amazon MQ mulai melacak perubahan pada kebijakan AWS terkelola. Untuk informasi selengkapnya, lihat berikut ini: • the section called “AWS kebijakan terkelola”
14 April 2021	Amazon MQ kini tersedia di Wilayah China (Beijing) dan China (Ningxia). Untuk informasi tentang wilayah yang tersedia, lihat Wilayah dan Titik Akhir AWS.
7 April 2021	Amazon MQ kini mendukung RabbitMQ 3.8.11. Untuk informasi selengkapnya tentang perbaikan dan fitur yang ada dalam pembaruan ini, lihat: • Catatan rilis RabbitMQ 3.8.11 pada repositori server RabbitMQ GitHub • RabbitMQ changelog • Mengelola Amazon MQ untuk versi mesin RabbitMQ
1 April 2021	Amazon MQ kini tersedia di Wilayah Asia Pacific (Osaka). Untuk informasi tentang wilayah yang tersedia, lihat Wilayah dan titik akhir Amazon MQ.
21 Desember 2020	Amazon MQ kini mendukung ActiveMQ 5.15.14. Untuk informasi lebih lanjut, lihat hal berikut: • Catatan Rilis ActiveMQ 5.15.14 • Mengelola versi mesin Amazon MQ for ActiveMQ • Menggunakan file konfigurasi Spring XML • **Important** Karena masalah Apache ActiveMQ yang diketahui dalam rilis ini, tombol Jeda Antrean di konsol web ActiveMQ tidak dapat digunakan dengan broker Amazon MQ for ActiveMQ. Untuk informasi selengkapnya tentang masalah ini, lihat AMQ-8104.

Date	Pembaruan Dokumentasi
4 November 2020	Amazon MQ kini mendukung RabbitMQ, broker pesan sumber terbuka populer. Ini memungkinkan Anda untuk memigrasikan broker pesan RabbitMQ yang ada ke AWS tanpa harus menulis ulang kode. Amazon MQ for RabbitMQ mengelola broker pesan individu dan terklaster serta menangani tugas seperti penyediaan infrastruktur, penyiapan broker, dan pembaruan perangkat lunak. • Amazon MQ mendukung RabbitMQ 3.8.6. Untuk informasi selengkapnya tentang versi mesin yang didukung, lihat the section called “Manajemen versi”. • Tingkat Gratis AWS menyertakan hingga 750 jam dari broker `mq.t3.micro` instans tunggal dan penyimpanan hingga 20 GB per bulan selama satu tahun. Untuk informasi selengkapnya tentang tipe instans yang didukung, lihat Broker instance types. • Dengan Amazon MQ untuk RabbitMQ, Anda dapat mengakses broker Anda menggunakan AMQP 0-9-1, dan dengan bahasa apa pun yang didukung oleh pustaka klien RabbitMQ. Untuk informasi selengkapnya tentang protokol dan cipher suite yang didukung, lihat the section called “Protokol Amazon MQ for RabbitMQ”. • Amazon MQ for RabbitMQ tersedia di semua wilayah tempat Amazon MQ tersedia saat ini. Untuk mempelajari selengkapnya tentang semua wilayah yang tersedia, lihat Tabel Wilayah AWS. Untuk mulai menggunakan Amazon MQ, membuat broker, dan menghubungkan aplikasi berbasis JVM ke broker RabbitMQ Anda, lihat Memulai: Membuat dan menghubungkan ke broker RabbitMQ.
22 Oktober 2020	Amazon MQ mendukung ActiveMQ 5.15.13. Untuk informasi lebih lanjut, lihat hal berikut: • Catatan Rilis ActiveMQ 5.15.13 • Mengelola versi mesin Amazon MQ for ActiveMQ • Menggunakan file konfigurasi Spring XML

Date	Pembaruan Dokumentasi
30 September 2020	Amazon MQ kini tersedia di Wilayah Europe (Milan). Untuk informasi tentang wilayah yang tersedia, lihat Wilayah dan titik akhir Amazon MQ.
27 Juli 2020	Anda dapat mengautentikasi pengguna Amazon MQ menggunakan kredensial yang disimpan di Direktori Aktif atau server LDAP lainnya. Anda juga dapat menambahkan, menghapus, dan memodifikasi pengguna Amazon MQ serta menetapkan izin untuk topik juga antrean. Untuk informasi selengkapnya, lihat Mengintegrasikan LDAP dengan ActiveMQ.
17 Juli 2020	Amazon MQ kini mendukung tipe instans `mq.t3.micro` . Untuk informasi selengkapnya, lihat Broker instance types.
30 Juni 2020	Amazon MQ mendukung ActiveMQ 5.15.12. Untuk informasi lebih lanjut, lihat hal berikut: • Catatan Rilis ActiveMQ 5.15.12 • Mengelola versi mesin Amazon MQ for ActiveMQ • Menggunakan file konfigurasi Spring XML
30 April 2020	Amazon MQ mendukung elemen pengumpulan anak baru, `systemUsage` , pada elemen `broker`. Untuk informasi selengkapnya, lihat systemUsage. Amazon MQ juga mendukung tiga atribut baru pada elemen anak `kahaDB`. • `journalDiskSyncInterval` - Interval (mdtk) untuk kapan harus melakukan sinkronisasi disk jika `journalDiskSyncStrategy=periodic` . • `journalDiskSyncStrategy` - mengonfigurasi kebijakan sinkronisasi disk. • `preallocationStrategy` - mengonfigurasi cara broker akan melakukan pra-alokasi file jurnal ketika file jurnal baru diperlukan. Untuk informasi selengkapnya, lihat Atribut.

Date	Pembaruan Dokumentasi
3 Maret 2020	Amazon MQ mendukung dua metrik baru CloudWatch • `TempPercentUsage` - Persentase penyimpanan sementara yang tersedia dan digunakan oleh pesan tidak tetap. • `JobSchedulerStorePercentUsage` - Persentase ruang disk yang digunakan oleh penyimpanan penjadwal tugas. Untuk informasi selengkapnya, lihat Monitoring and logging Amazon MQ brokers.
4 Februari 2020	Amazon MQ tersedia di wilayah Asia Pacific (Hong Kong) dan Middle East (Bahrain). Untuk informasi tentang wilayah yang tersedia, lihat Wilayah dan Titik Akhir AWS.
22 Januari 2020	Amazon MQ mendukung ActiveMQ 5.15.10. Untuk informasi lebih lanjut, lihat hal berikut: • Catatan Rilis ActiveMQ 5.15.10 • Mengelola versi mesin Amazon MQ for ActiveMQ • Menggunakan file konfigurasi Spring XML
19 Desember 2019	Amazon MQ tersedia di wilayah Europe (Stockholm) dan South America (São Paulo). Untuk informasi tentang wilayah yang tersedia, lihat Wilayah dan Titik Akhir AWS.

Date	Pembaruan Dokumentasi
16 Desember 2019	Amazon MQ mendukung pembuatan broker yang dioptimalkan throughput menggunakan Amazon Elastic Block Store (EBS)—bukan Amazon Elastic File System (Amazon EFS) default—untuk penyimpanan broker. Untuk memanfaatkan daya tahan dan replikasi yang tinggi di beberapa Availabil ity Zone, gunakan Amazon EFS. Untuk memanfaatkan latensi rendah dan throughput yang tinggi, gunakan Amazon EBS. **Important** • Anda dapat menggunakan Amazon EBS hanya dengan keluarga tipe instans broker `mq.m5`. • Meski Anda dapat mengubah tipe instans broker, Anda tidak dapat mengubah tipe penyimpanan broker setelah Anda membuat broker. • Amazon EBS mereplikasi data dalam satu Availability Zone dan tidak mendukung mode deployment ActiveMQ aktif/siaga. Untuk informasi lebih lanjut, lihat hal berikut: • Storage • Pilih jenis penyimpanan broker yang tepat untuk throughput terbaik • Properti `storageType` dari sumber daya `broker-instance-options` dalam Referensi REST API Amazon MQ • Metrik `BurstBalance`, `VolumeReadOps`, dan `VolumeWriteOps` di bagian Monitoring and logging Amazon MQ brokers.
18 Oktober 2019	Dua CloudWatch metrik Amazon tersedia: `TotalEnqueueCount` dan`TotalDequeueCount`. Untuk informasi lebih lanjut, lihat Monitoring and logging Amazon MQ brokers

Date	Pembaruan Dokumentasi
11 Oktober 2019	Amazon MQ kini mendukung titik akhir kepatuhan Standar Pemrosesan Informasi Federal 140-2 (FIPS) di wilayah komersial AS. Untuk informasi lebih lanjut, lihat hal berikut: • Standar Pemrosesan Informasi Federal (FIPS) 140-2 • Wilayah dan Titik Akhir Amazon MQ
30 September 2019	Amazon MQ kini menyertakan kemampuan untuk menskalakan broker Anda dengan mengubah tipe instans host. Untuk informasi selengkapnya, lihat properti `hostInstanceType` dari `UpdateBrokerInput`, dan properti `pendingHostInstanceType` dari `DescribeBrokerOutput`.
30 Agustus 2019	Kini Anda dapat memperbarui grup keamanan yang terkait dengan broker, baik di konsol maupun dengan `UpdateBrokerInput`.
22 Juli 2019	Amazon MQ terintegrasi dengan AWS Key Management Service (KMS) untuk menawarkan enkripsi sisi server. Sekarang Anda dapat memilih CMK yang dikelola pelanggan Anda sendiri, atau menggunakan kunci KMS AWS terkelola di akun Anda AWS KMS . Untuk informasi selengkapnya, lihat Enkripsi saat diam. Amazon MQ mendukung penggunaan AWS KMS kunci dengan cara berikut. • AWS kunci KMS yang dimiliki - Kuncinya dimiliki Amazon MQ dan tidak ada di akun Anda. • AWS kunci KMS AWS terkelola - Kunci KMS terkelola (aws/mq) adalah kunci KMS di akun Anda yang dibuat, dikelola, dan digunakan atas nama Anda oleh Amazon MQ. • Pilih CMK terkelola pelanggan yang ada — Pelanggan CMKs yang dikelola dibuat dan dikelola oleh Anda di AWS Key Management Service (KMS).
19 Juni 2019	Amazon MQ tersedia di wilayah Europe (Paris) dan Asia Pacific (Mumbai). Untuk informasi tentang wilayah yang tersedia, lihat Wilayah dan Titik Akhir AWS.

Date	Pembaruan Dokumentasi
12 Juni 2019	Amazon MQ tersedia di wilayah Canada (Central). Untuk informasi tentang wilayah yang tersedia, lihat Wilayah dan Titik Akhir AWS.
3 Juni 2019	Dua CloudWatch metrik Amazon baru tersedia: `EstablishedConnectionsCount` dan `InactiveDurableSubscribers`. Untuk informasi selengkapnya, lihat berikut ini: • Monitoring and logging Amazon MQ brokers • Monitoring and logging Amazon MQ brokers
10 Mei 2019	Penyimpanan data untuk tipe instans `mq.t2.micro` terbatas pada 20 GB. Untuk informasi lebih lanjut, lihat hal berikut: • the section called “Penyimpanan Data” • Broker instance types
29 April 2019	Kini Anda dapat menggunakan kebijakan berbasis tanda dan izin tingkat sumber daya. Untuk informasi lebih lanjut, lihat hal berikut: • Cara kerja Amazon MQ dengan IAM • Izin tingkat sumber daya untuk tindakan API Amazon MQ
16 April 2019	Kini Anda dapat mengambil informasi tentang mesin broker dan opsi instans broker menggunakan REST API. Untuk informasi lebih lanjut, lihat hal berikut: • Opsi instans broker • Jenis mesin broker
8 April 2019	Amazon MQ mendukung ActiveMQ 5.15.9. Untuk informasi lebih lanjut, lihat hal berikut: • Catatan Rilis ActiveMQ 5.15.9 • Mengelola versi mesin Amazon MQ for ActiveMQ • Menggunakan file konfigurasi Spring XML

Date	Pembaruan Dokumentasi
4 Maret 2019	Peningkatan dokumentasi untuk mengonfigurasi failover dinamis dan menyeimbangkan ulang klien untuk jaringan broker. Aktifkan failover dinamis dengan mengonfigurasi `transportConnectors` bersama dengan opsi konfigurasi `networkConnectors` . Untuk informasi lebih lanjut, lihat hal berikut: • Failover Dinamis dengan Konektor Transportasi • Jaringan broker Amazon MQ • Amazon MQ Broker Configuration Parameters
27 Februari 2019	Amazon MQ tersedia di Wilayah Europe (London) selain wilayah berikut: • Asia Pacific (Singapore) • US East (Ohio) • AS Timur (Virginia Utara) • AS Barat (California Utara) • US West (Oregon) • Asia Pacific (Tokyo) • Asia Pacific (Seoul) • Asia Pacific (Sydney) • Europe (Frankfurt) • Europe (Ireland)
24 Januari 2019	Konfigurasi default kini menyertakan kebijakan untuk membersihkan tujuan yang tidak aktif.
17 Januari 2019	Tipe instans `mq.t2.micro` Amazon MQ kini hanya mendukung 100 koneksi per protokol tingkat wire. Untuk informasi selengkapnya, lihat, Quotas in Amazon MQ.

Date	Pembaruan Dokumentasi
19 Desember 2018	Anda dapat mengonfigurasi serangkaian broker Amazon MQ dalam jaringan broker. Untuk informasi selengkapnya, lihat bagian berikut: • Jaringan broker Amazon MQ • Creating and Configuring a Network of Brokers • Mengonfigurasi Jaringan Broker dengan Benar • `networkConnector` • `networkConnectionStartAsinkron`

11 Desember 2018	Amazon MQ mendukung ActiveMQ 5.15.8, 5.15.6, dan 5.15.0.
	• Bug yang diselesaikan dan peningkatan di ActiveMQ:
	  • Catatan Rilis ActiveMQ 5.15.8
	  • Catatan Rilis ActiveMQ 5.15.7

5 Desember 2018	AWS mendukung penandaan sumber daya untuk membantu melacak alokasi biaya Anda. Anda dapat menandai sumber daya saat membuatnya, atau dengan melihat detail sumber daya tersebut. Untuk informasi selengkapnya, lihat Menandai sumber daya.
19 November 2018	AWS telah memperluas program kepatuhan SOC untuk memasukkan Amazon MQ sebagai layanan yang sesuai dengan SOC.
15 Oktober 2018	• Jumlah maksimum grup per pengguna adalah 20. Untuk informasi selengkapnya, lihat Pengguna. • Jumlah maksimum koneksi per broker, per protokol tingkat wire adalah 1.000. Untuk informasi selengkapnya, lihat Pialang.
2 Oktober 2018	AWS telah memperluas program kepatuhan HIPAA untuk memasukkan Amazon MQ sebagai Layanan yang Memenuhi Syarat HIPAA.

Date	Pembaruan Dokumentasi

27 September 2018	Amazon MQ mendukung ActiveMQ 5.15.6, selain 5.15.0. Untuk informasi lebih lanjut, lihat hal berikut:
	• Memulai: Membuat dan menghubungkan ke broker ActiveMQ
	• Bug yang diselesaikan dan peningkatan dalam dokumentasi ActiveMQ:
	  • Catatan Rilis ActiveMQ 5.15.6
	  • Catatan Rilis ActiveMQ 5.15.5
	  • Catatan Rilis ActiveMQ 5.15.4
	  • Catatan Rilis ActiveMQ 5.15.3
	  • Catatan Rilis ActiveMQ 5.15.2
	  • Catatan Rilis ActiveMQ 5.15.1
	• Klien ActiveMQ 5.15.6

31 Agustus 2018	• Metrik berikut tersedia:
	  • `CurrentConnectionsCount`
	  • `TotalConsumerCount`
	  • `TotalProducerCount`
	  Untuk informasi selengkapnya, lihat bagian Monitoring and logging Amazon MQ brokers.
	• Alamat IP broker ditampilkan pada halaman Detail.
	  Note
	  Untuk broker dengan aksesibilitas publik yang dinonaktifkan, alamat IP internal ditampilkan.

Date	Pembaruan Dokumentasi
30 Agustus 2018	Amazon MQ tersedia di Wilayah Asia Pacific (Singapore) selain wilayah berikut: • US East (Ohio) • AS Timur (Virginia Utara) • AS Barat (California Utara) • US West (Oregon) • Asia Pacific (Tokyo) • Asia Pacific (Seoul) • Asia Pacific (Sydney) • Europe (Frankfurt) • Europe (Ireland)
30 Juli 2018	Anda dapat mengonfigurasi Amazon MQ untuk mempublikasikan log umum dan audit ke Amazon CloudWatch Logs. Untuk informasi selengkapnya, lihat Monitoring and logging Amazon MQ brokers.
25 Juli 2018	Amazon MQ tersedia di Wilayah Asia Pacific (Tokyo) dan Asia Pacific (Seoul) selain wilayah berikut: • US East (Ohio) • AS Timur (Virginia Utara) • AS Barat (California Utara) • US West (Oregon) • Asia Pacific (Sydney) • Europe (Frankfurt) • Europe (Ireland)
19 Juli 2018	Anda dapat menggunakan AWS CloudTrail untuk mencatat panggilan API Amazon MQ. Untuk informasi selengkapnya, lihat Logging Amazon MQ API calls using CloudTrail.

Date	Pembaruan Dokumentasi
29 Juni 2018	Selain `mq.t2.micro` dan `mq.m4.large`, tipe instans broker berikut tersedia untuk pengembangan, pengujian, dan beban kerja produksi reguler yang memerlukan throughput tinggi: • `mq.m5.large` • `mq.m5.xlarge` • `mq.m5.2xlarge` • `mq.m5.4xlarge` Untuk informasi selengkapnya, lihat Broker instance types.
27 Juni 2018	Amazon MQ tersedia di Wilayah US West (N. California) selain wilayah berikut: • US East (Ohio) • AS Timur (Virginia Utara) • US West (Oregon) • Asia Pacific (Sydney) • Europe (Frankfurt) • Europe (Ireland)

Date	Pembaruan Dokumentasi

14 Juni 2018	• Anda dapat menggunakan `AWS::Amazon MQ::Broker` AWS CloudFormation sumber daya untuk melakukan tindakan berikut:
	  • Membuat broker.
	  • Menambahkan perubahan konfigurasi atau memodifikasi pengguna untuk broker yang ditentukan.
	  • Mengembalikan informasi tentang broker yang ditentukan.
	  • Menghapus broker yang ditentukan.
	
	  Note
	  Saat Anda mengubah properti apa pun dari jenis properti Broker Amazon MQ Broker ConfigurationId atau Amazon MQ Broker User, broker segera di-boot ulang.
	• Anda dapat menggunakan `AWS::Amazon MQ::Configuration` AWS CloudFormation sumber daya untuk melakukan tindakan berikut:
	  • Membuat konfigurasi.
	  • Memperbarui konfigurasi yang ditentukan.
	  • Mengembalikan informasi tentang konfigurasi yang ditentukan.
	
	  Note
	  Anda dapat menggunakan CloudFormation untuk memodifikasi—tetapi tidak menghapus—konfigurasi Amazon MQ.

7 Juni 2018	Konsol Amazon MQ mendukung bahasa Jerman, Portugis Brasil, Spanyol, Italia, dan Tionghoa Tradisional.
17 Mei 2018	Batas jumlah pengguna per broker adalah 250. Untuk informasi selengkap nya, lihat Pengguna.
13 Maret 2018	Pembuatan broker berlangsung sekitar 15 menit. Untuk informasi selengkap nya, lihat Selesai membuat broker.

Date	Pembaruan Dokumentasi
1 Maret 2018	• Anda dapat menggunakan penyimpanan dan pengiriman bersamaan untuk Apache KahaDB menggunakan atribut `concurrentStoreAndDispatchQueues`. • `CpuCreditBalance` CloudWatch Metrik > tersedia untuk jenis instans `mq.t2.micro` broker.
10 Januari 2018	Perubahan berikut memengaruhi konsol Amazon MQ: • Dalam daftar broker, kolom pembuatan tersembunyi secara default. Untuk menyesuaikan ukuran halaman dan kolom, pilih  . • Pada ***MyBroker***halaman, di bagian Koneksi, pilih nama grup keamanan Anda atau  buka konsol EC2 (bukan konsol VPC). Konsol EC2 memungkinkan konfigurasi aturan masuk dan keluar yang lebih intuitif. Untuk informasi selengkapnya, lihat bagian Connecting a Java application to your broker yang diperbarui.
9 Januari 2018	• Izin untuk ID operasi REST `UpdateBroker` tercantum dengan benar sebagai `mq:UpdateBroker` pada konsol IAM. • Izin `mq:DescribeEngine` yang salah dihapus dari konsol IAM.

Date	Pembaruan Dokumentasi

28 November 2017	Ini adalah rilis awal dari Amazon MQ dan Rilis Developer Amazon MQ.
	• Amazon MQ tersedia di wilayah berikut:
	  • US East (Ohio)
	  • AS Timur (Virginia Utara)
	  • US West (Oregon)
	  • Asia Pacific (Sydney)
	  • Europe (Frankfurt)
	  • Europe (Ireland)
	  Penggunaan tipe instans `mq.t2.micro` tunduk pada kredit CPU dan performa dasar—dengan kemampuan untuk burst di atas tingkat dasar (untuk informasi selengkapnya, lihat metrik `CpuCreditBalance`). Jika aplikasi Anda membutuhkan performa tetap, pertimbangkan untuk menggunakan tipe instans `mq.m5.large`.
	• Anda dapat membuat broker `mq.m4.large` dan `mq.t2.micro`.
	  Penggunaan tipe instans `mq.t2.micro` tunduk pada kredit CPU dan performa dasar—dengan kemampuan untuk burst di atas tingkat dasar (untuk informasi selengkapnya, lihat metrik `CpuCreditBalance`). Jika aplikasi Anda membutuhkan performa tetap, pertimbangkan untuk menggunakan tipe instans `mq.m5.large`.
	• Anda dapat menggunakan mesin broker ActiveMQ 5.15.0.
	• Anda juga dapat membuat dan mengelola broker secara terprogram menggunakan Amazon MQ REST API dan. AWS SDKs
	• Anda dapat mengakses broker menggunakan bahasa pemrograman yang didukung ActiveMQ dan dengan mengaktifkan TLS secara eksplisit untuk protokol berikut:
	  • AMQP
	  • MQTT
	  • MQTT lebih WebSocket
	  • OpenWire

Date	Pembaruan Dokumentasi
	• MENGINJAK • STOMP berakhir WebSocket • Anda dapat terhubung ke broker ActiveMQ menggunakan berbagai klien ActiveMQ. Kami merekomendasikan penggunaan Klien ActiveMQ. Untuk informasi selengkapnya, lihat Connecting a Java application to your broker. • Broker Anda dapat mengirim dan menerima pesan dengan berbagai ukuran.

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