



Panduan Developer

# Amazon MQ



# Amazon MQ: Panduan Developer

Copyright © 2024 Amazon Web Services, Inc. and/or its affiliates. All rights reserved.

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

---

# Table of Contents

Apa itu Amazon MQ? .....	1
Apa saja perbedaan Amazon MQ dengan Amazon SQS atau Amazon SNS? .....	1
Bagaimana cara memulai dengan Amazon MQ? .....	1
Kami ingin mendengar pendapat Anda .....	2
Menyiapkan .....	3
Langkah 1: Prasyarat .....	3
Daftar Akun AWS .....	3
Membuat pengguna administratif .....	4
Buat pengguna dan dapatkan AWS kredensialmu .....	5
Langkah 3: bersiap menggunakan kode contoh .....	6
Langkah selanjutnya .....	6
Memulai .....	8
Prasyarat .....	8
Membuat dan menghubungkan ke broker ActiveMQ .....	8
Langkah 1: membuat broker ActiveMQ .....	9
Langkah 2: menghubungkan aplikasi Java ke broker Anda .....	11
Langkah 3: (Opsional) terhubung keAWS Lambdafungsi .....	16
Langkah 4: hapus broker Anda .....	19
Langkah selanjutnya .....	19
Membuat dan menghubungkan ke broker RabbitMQ .....	19
Langkah 1: membuat broker RabbitMQ .....	20
Langkah 2: menghubungkan aplikasi berbasis JVM ke broker Anda .....	22
Langkah 3: (Opsional) terhubung ke suatu AWS Lambda fungsi .....	26
Langkah 4: hapus broker Anda .....	29
Langkah selanjutnya .....	30
Mengelola broker .....	31
Mempertahankan broker .....	31
Menyesuaikan jendela pemeliharaan broker .....	32
Meningkatkan versi mesin .....	35
Meningkatkan versi mesin secara manual .....	36
Otomatis meningkatkan versi mesin minor .....	39
Status broker .....	40
Membuat daftar broker dan melihat detail broker .....	41
Untuk membuat daftar broker dan melihat detail broker .....	41

Mengakses konsol web broker tanpa aksesibilitas publik .....	44
Prasyarat .....	44
Untuk mengakses konsol web broker pada Broker tanpa Aksesibilitas Publik .....	45
Melakukan boot ulang broker .....	46
Untuk Melakukan Boot Ulang Broker Amazon MQ .....	46
Menghapus broker .....	47
Menghapus broker Amazon MQ .....	47
Konfigurasi broker .....	47
Siklus hidup konfigurasi broker .....	47
Tipe instans .....	48
Tipe instans Amazon MQ for ActiveMQ .....	49
Tipe instans Amazon MQ for RabbitMQ .....	55
Penandaan sumber daya .....	57
Penandaan untuk Alokasi Biaya .....	57
Mengelola Tanda di Konsol Amazon MQ .....	58
Mengelola Menggunakan Tindakan API Amazon MQ .....	60
Amazon MQ for ActiveMQ .....	61
Mesin ActiveMQ .....	61
Elemen Dasar .....	62
Arsitektur broker .....	75
Konfigurasi broker .....	89
Manajemen versi .....	124
Contoh kerja Java .....	127
Tutorial ActiveMQ .....	138
Membuat dan mengonfigurasi broker .....	138
Membuat dan mengonfigurasi jaringan broker .....	145
Menghubungkan aplikasi Java ke broker Anda .....	151
Mengintegrasikan broker ActiveMQ dengan LDAP .....	156
Membuat dan mengelola pengguna broker .....	172
Praktik terbaik Amazon MQ for ActiveMQ .....	175
Terhubung ke Amazon MQ .....	175
Memastikan performa Amazon MQ yang efektif .....	178
Hindari mulai ulang lambat dengan memulihkan transaksi XA yang disiapkan .....	181
Replikasi data lintas wilayah .....	182
Pialang primer dan replika .....	183
Membuat/menghapus broker CRDR .....	184

Memulai switchover/failover .....	188
Metrik .....	191
Kuota .....	193
Pialang .....	193
Konfigurasi .....	194
Pengguna .....	195
Penyimpanan Data .....	196
Throttling API .....	197
Amazon MQ for RabbitMQ .....	198
Mesin RabbitMQ .....	198
Elemen dasar .....	198
Arsitektur broker .....	217
Konfigurasi broker .....	220
Manajemen versi .....	225
Tutorial RabbitMQ .....	228
Mengedit preferensi broker .....	228
Menggunakan Python Pika dengan Amazon MQ for RabbitMQ .....	229
Menyelesaikan sinkronisasi antrean yang dijeda .....	236
Amazon MQ untuk praktik terbaik RabbitMQ .....	242
Mengaktifkan antrean malas .....	243
Gunakan antrean yang persisten dan tahan lama .....	244
Jaga antrian pendek .....	245
Konfigurasikan pengakuan dan konfirmasi .....	246
Konfigurasikan pra-pengambilan .....	247
Konfigurasikan Seledri .....	248
Secara otomatis pulih dari kegagalan jaringan .....	248
Aktifkan Classic Queue v2 untuk broker RabbitMQ Anda .....	249
Quotas .....	250
Pialang .....	250
Penyimpanan Data .....	251
Throttling API .....	251
Keamanan .....	253
Perlindungan data .....	254
Enkripsi .....	255
Enkripsi saat istirahat .....	255
Enkripsi dalam transit .....	264

Identity and access management .....	266
Penonton .....	267
Mengautentikasi menggunakan identitas .....	267
Mengelola akses menggunakan kebijakan .....	271
Cara kerja Amazon MQ dengan IAM .....	273
Contoh kebijakan berbasis identitas .....	279
Autentikasi dan otorisasi API .....	282
Kebijakan terkelola AWS .....	287
Menggunakan peran yang terhubung dengan layanan .....	288
Pemecahan Masalah .....	294
Validasi Kepatuhan .....	296
Ketahanan .....	297
Keamanan infrastruktur .....	297
Praktik terbaik keamanan .....	298
Lebih memilih broker tanpa aksesibilitas publik .....	298
Selalu konfigurasi peta otorisasi .....	299
Memblokir Protokol yang Tidak Diperlukan .....	299
Pencatatan dan pemantauan .....	301
Mengakses metrik CloudWatch .....	301
AWS Management Console .....	302
AWS Command Line Interface .....	304
CloudWatchAPI Amazon .....	304
Pemantauan broker menggunakan CloudWatch .....	304
Pencatatan dan pemantauan Amazon MQ untuk broker ActiveMQ .....	305
Pencatatan dan pemantauan Amazon MQ untuk broker RabbitMQ .....	314
Pembuatan log panggilan API CloudTrail .....	321
Informasi Amazon MQ di CloudTrail .....	322
Contoh Entri Berkas Log Amazon MQ .....	324
Mengonfigurasi Amazon MQ untuk memublikasikan CloudWatch log ke Logs .....	326
Mengonfigurasi Amazon MQ untuk ActiveMQ .....	326
Mengonfigurasi Amazon MQ untuk RabbitMQ .....	332
Kuota .....	333
Pialang .....	333
Konfigurasi .....	334
Pengguna .....	335
Penyimpanan Data .....	336

Throttling API .....	337
Pemecahan Masalah .....	339
Pemecahan masalah: Umum .....	340
Saya tidak dapat terhubung ke konsol web broker atau titik akhir saya. ....	340
Pengecualian SSL .....	346
Saya membuat broker tetapi pembuatan broker gagal. ....	346
Broker saya memulai kembali dan saya tidak yakin mengapa. ....	346
Pemecahan masalah: Amazon MQ untuk ActiveMQ .....	347
Mengambil CloudWatch Log .....	348
Menghubungkan ke broker setelah restart .....	348
Beberapa klien tidak dapat terhubung .....	349
JSPengecualian di konsol web .....	350
Pemecahan masalah: Amazon MQ untuk RabbitMQ .....	350
Saya tidak dapat melihat metrik untuk antrian atau host virtual saya di CloudWatch .....	351
Bagaimana cara mengaktifkan plugin di Amazon MQ untuk RabbitMQ? .....	351
Saya tidak dapat mengubah konfigurasi VPC Amazon untuk broker. ....	351
Pemecahan masalah: Kode yang diperlukan tindakan Amazon MQ .....	351
RABBITMQ_MEMORY_ALARM .....	352
RABBITMQ_INVALID_KMS_KEY .....	359
BROKER_ENI_DELETED .....	360
BROKER_OOM .....	361
RABBITMQ_DISK_ALARM .....	362
Sumber daya terkait .....	365
Sumber daya Amazon MQ .....	365
Sumber daya Amazon MQ for ActiveMQ .....	366
Sumber daya Amazon MQ for RabbitMQ .....	366
Catatan rilis .....	368
Riwayat Dokumen .....	397
AWSGlosarium .....	412
.....	cdxiii

# Apa itu Amazon MQ?

Amazon MQ adalah layanan broker pesan terkelola yang memudahkan migrasi ke broker pesan di cloud. Broker pesan memungkinkan aplikasi dan komponen perangkat lunak untuk berkomunikasi menggunakan berbagai bahasa pemrograman, sistem operasi, dan protokol olahpesan formal. [Saat ini, Amazon MQ mendukung jenis mesin Apache ActiveMQ Classic dan RabbitMQ.](#)

Amazon MQ bekerja dengan aplikasi dan layanan yang ada tanpa perlu mengelola, mengoperasikan, atau mempertahankan sistem olahpesan milik Anda.

## Topik

- [Apa saja perbedaan Amazon MQ dengan Amazon SQS atau Amazon SNS?](#)
- [Bagaimana cara memulai dengan Amazon MQ?](#)
- [Kami ingin mendengar pendapat Anda](#)

# Apa saja perbedaan Amazon MQ dengan Amazon SQS atau Amazon SNS?

Amazon MQ adalah layanan broker pesan terkelola yang menyediakan kompatibilitas dengan banyak broker pesan populer. Kami merekomendasikan Amazon MQ untuk memigrasikan aplikasi dari broker pesan yang ada yang mengandalkan kompatibilitas dengan API seperti JMS atau protokol seperti AMQP 0-9-1, AMQP 1.0, MQTT, dan STOMP. OpenWire

[Amazon SQS](#) dan [Amazon SNS](#) adalah layanan antrean dan topik yang sangat dapat diskalakan, mudah digunakan, dan tidak mengharuskan Anda menyiapkan broker pesan. Kami merekomendasikan layanan ini untuk aplikasi baru yang dapat memanfaatkan skalabilitas yang hampir tak terbatas dan API sederhana.

# Bagaimana cara memulai dengan Amazon MQ?

- Untuk membuat broker pertama Anda dengan Amazon MQ, lihat [Getting Started with Amazon MQ](#).
- Untuk mengetahui pedoman dan peringatan yang akan membantu Anda memaksimalkan Amazon MQ, lihat dan [Working with Amazon MQ for ActiveMQ](#) [Working with Amazon MQ for RabbitMQ](#)
- Untuk mempelajari tentang REST API Amazon MQ, lihat [Referensi REST API Amazon MQ](#).



- Untuk mempelajari tentang AWS CLI perintah Amazon MQ, lihat [Amazon MQ di AWS CLI Referensi Perintah](#).

## Kami ingin mendengar pendapat Anda

Kami menyambut umpan balik Anda. Untuk menghubungi kami, kunjungi [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

### Daftar Akun AWS

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

Untuk mendaftar Akun AWS

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

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

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

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

## Membuat pengguna administratif

Setelah Anda mendaftarkan 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.

### Mengamankan Pengguna root akun AWS Anda

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

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

2. Mengaktifkan autentikasi multi-faktor (MFA) untuk pengguna root Anda.

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

### Membuat pengguna administratif

1. Aktifkan Pusat Identitas IAM.

Untuk petunjuk, lihat [Mengaktifkan AWS IAM Identity Center](#) di Panduan AWS IAM Identity Center Pengguna.

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

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

### Masuk sebagai pengguna administratif

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

Untuk bantuan masuk menggunakan pengguna Pusat Identitas IAM, silakan lihat [Masuk ke portal akses AWS](#) dalam Panduan Pengguna AWS Sign-In.

## Langkah 2: buat pengguna dan dapatkan AWS kredensialnya

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

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

Pengguna mana yang membutuhkan akses terprogram?	Ke	Oleh
Identitas tenaga kerja  (Pengguna yang dikelola di Pusat Identitas IAM)	Gunakan kredensial sementara untuk menandatangani permintaan terprogram ke AWS CLI, SDK AWS, atau API AWS.	Mengikuti petunjuk untuk antarmuka yang ingin Anda gunakan. <ul style="list-style-type: none"> <li>• Untuk AWS CLI, lihat <a href="#">Mengonfigurasi AWS CLI untuk menggunakan AWS IAM Identity Center</a> di Panduan Pengguna AWS Command Line Interface.</li> <li>• Untuk SDK AWS, alat, dan API AWS, lihat <a href="#">Autentikasi Pusat Identitas IAM</a> di Panduan Referensi SDK dan Alat AWS.</li> </ul>
IAM	Gunakan kredensial sementara untuk menandatangani permintaan terprogram ke AWS CLI, SDK AWS, atau API AWS.	Ikuti petunjuk dalam <a href="#">Menggunakan kredensial sementara dengan sumber daya AWS</a> di Panduan Pengguna IAM.
IAM	(Tidak disarankan) Gunakan kredensial jangka panjang untuk menandatangani permintaan terprogram	Mengikuti petunjuk untuk antarmuka yang ingin Anda gunakan.

Pengguna mana yang membutuhkan akses terprogram?	Ke	Oleh
	ke AWS CLI, AWS SDK, atau API AWS.	<ul style="list-style-type: none"> <li>• Untuk AWS CLI, lihat <a href="#">Mengautentikasi menggunakan kredensial pengguna IAM</a> di Panduan Pengguna AWS Command Line Interface.</li> <li>• Untuk SDK dan alat AWS, lihat <a href="#">Mengautentikasi menggunakan kredensial jangka panjang</a> di Panduan Referensi SDK dan Alat AWS.</li> <li>• Untuk API AWS, lihat <a href="#">Mengelola kunci akses untuk pengguna IAM</a> di Panduan Pengguna IAM.</li> </ul>

## Langkah 3: bersiap menggunakan kode contoh

Tutorial berikut menunjukkan cara bekerja dengan broker Amazon MQ menggunakan AWS Management Console serta cara terhubung ke broker Amazon MQ for ActiveMQ dan Amazon MQ for 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 [REST API](#) Amazon MQ dan SDK AWS.

## 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 dengan Amazon MQ

Bagian ini akan membantu Anda menjadi lebih familier dengan Amazon MQ dengan menunjukkan cara membuat Amazon MQ untuk broker ActiveMQ atau RabbitMQ dan cara menghubungkan aplikasi Anda.

Membuat dan menghubungkan ke instans broker sedikit berbeda untuk setiap mesin broker. Pilih salah satu jenis mesin berikut yang ingin Anda gunakan untuk informasi detail tentang membuat dan menghubungkan ke broker. Setelah membuat dan terhubung ke broker, Anda dapat menemukan petunjuk untuk membantu menghapusnya.

Topik

- [Prasyarat](#)
- [Membuat dan menghubungkan ke broker ActiveMQ](#)
- [Membuat dan menghubungkan ke broker RabbitMQ](#)

## Prasyarat

Sebelum memulai, selesaikan langkah-langkah di [Setting Up Amazon MQ](#).

## 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 instans broker kelas (m5, t3) dan ukuran (large, micro) adalah tipe instans broker (misalnya, mq.m5.large). Untuk informasi selengkapnya, lihat [Pialang](#).

Topik

- [Langkah 1: membuat broker ActiveMQ](#)
- [Langkah 2: menghubungkan aplikasi Java ke broker Anda](#)
- [Langkah 3: \(Opsional\) terhubung keAWS Lambdafungsi](#)
- [Langkah 4: hapus broker Anda](#)
- [Langkah selanjutnya](#)

## Langkah 1: membuat broker ActiveMQ

Tugas Amazon MQ yang pertama dan paling umum adalah membuat broker. Contoh berikut menunjukkan cara menggunakan AWS Management Console 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 [Broker Architecture](#).
    - SEBUAHPialang satu contohterdiri 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 [Broker 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 [Broker aktif/siaga Amazon MQ untuk ketersediaan tinggi](#).
    - Untuk informasi selengkapnya tentang cetak biru sampel jaringan broker, lihat [Cetak biru sampel](#).
  - b. Pilih Jenis penyimpanan (misalnya, EBS). Untuk informasi selengkapnya, lihat [Storage](#).
- c. Pilih Selanjutnya.
4. Di halaman Konfigurasi pengaturan, pada bagian Detail, lakukan hal berikut:
  - a. Masukkan nama Broker.

### Note

Amazon EBS mereplikasi data dalam satu Availability Zone dan tidak mendukung mode deployment [ActiveMQ aktif/siaga](#).



**⚠ Important**

Jangan menambahkan informasi identitas pribadi (PII) atau informasi rahasia atau sensitif lainnya dalam nama broker. Nama broker dapat diakses oleh orang lainAWSlayanan, termasukCloudWatchLog. 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](#).
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 orang lainAWSlayanan, termasukCloudWatchLog. 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.

	Name ▼	Status ▼	Deployment mode ▼	Instance type ▼
<input type="radio"/>	MyBroker	Running	Single-instance broker	mq.m5.large

## 7. Pilih **MyBroker**.

Pada **MyBroker** halaman, diHubungkan bagian, perhatikan broker Anda [Konsol web ActiveMQ URL](#), misalnya:

```
https://b-1234a5b6-78cd-901e-2fgh-3i45j6k17819-1.mq.us-east-2.amazonaws.com:8162
```

Perhatikan juga [Titik Akhir protokol tingkat wire](#). Berikut ini adalah contoh dari sebuah OpenWire titik akhir:

```
ssl://b-1234a5b6-78cd-901e-2fgh-3i45j6k17819-1.mq.us-east-2.amazonaws.com:61617
```

## Langkah 2: menghubungkan aplikasi Java ke broker 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 antrian, 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](#).

### Prasyarat

#### Mengaktifkan atribut VPC


##### Note

Anda tidak dapat menonaktifkan aksesibilitas publik untuk broker Amazon MQ Anda yang ada.

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, diKoneksibagian, perhatikan 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>5.15.8</version>
  </dependency>
  <dependency>
    <groupId>org.apache.activemq</groupId>
    <artifactId>activemq-pool</artifactId>
    <version>5.15.8</version>
  </dependency>
</dependencies>
```

```
</dependencies>
```

Untuk informasi selengkapnya tentang `activemq-client.jar`, lihat [Konfigurasi Awal](#) dalam 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.

## Membuat produsen pesan dan mengirimkan pesan

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

#### Note

Untuk broker aktif/siaga, Amazon MQ menyediakan dua URL Konsol Web ActiveMQ, tetapi hanya satu URL 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 [Broker Architecture](#).

Untuk titik akhir protokol tingkat wire, Anda dapat mengizinkan aplikasi untuk terhubung ke titik akhir menggunakan [Transportasi Failover](#).

```
// 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, antrian 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 produser.

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

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

## Membuat konsumen 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, antrian 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 olahpesan AWS (seperti Amazon SQS), konsumen terus-menerus terhubung ke broker.

#### 4. Menutup konsumen, sesi, dan koneksi.

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

## Langkah 3: (Opsional) terhubung keAWS Lambdafungsi


AWS Lambda dapat terhubung ke dan mengonsumsi pesan dari broker Amazon MQ Anda. Saat Anda menghubungkan broker ke Lambda, Anda membuat [pemetaan sumber acara](#) yang membaca pesan dari antrian dan memanggil fungsi [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 fungsi Lambda Anda [peran eksekusi](#).


- [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:
  - Konfigurasi satu NAT gateway per subnet publik. Untuk informasi lebih lanjut, lihat [Akses internet dan layanan untuk fungsi yang terhubung dengan VPC](#) di AWS Lambda Panduan Pengembang.
  - 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 lebih lanjut, lihat [Mengkonfigurasi titik akhir VPC antarmuka untuk Lambda](#) di AWS Lambda Panduan Pengembang.
3. [Konfigurasi broker Anda sebagai sumber acara](#) untuk fungsi Lambda menggunakan AWS Management Console. 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 acara 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": "QUJD0kFBQUE=",
      "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": "3DT00W7crj51prgVLQaGQ82S48k=",
      "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 AWS Lambda Panduan Pengembang.

## Langkah 4: hapus broker Anda

Jika Anda tidak menggunakan broker Amazon MQ (dan tidak akan menggunakannya dalam waktu dekat), praktik terbaiknya adalah menghapus broker dari Amazon MQ untuk mengurangi biaya AWS.

Contoh berikut menunjukkan cara menghapus broker menggunakan AWS Management Console.

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.

## Langkah selanjutnya

Sekarang Anda telah membuat broker, menghubungkan aplikasi ke broker, serta mengirim dan menerima pesan, Anda mungkin ingin mencoba hal berikut:

- [Creating and configuring a broker](#) (Pengaturan Tambahan)
- [Mengedit versi mesin broker, jenis instans, CloudWatch log, dan preferensi pemeliharaan](#)
- [Creating and applying broker configurations](#)
- [Listing brokers and viewing broker details](#)
- [Membuat dan mengelola pengguna broker ActiveMQ](#)
- [Rebooting a Broker](#)
- [Mengakses CloudWatch metrik untuk Amazon MQ](#)

Anda juga bisa mulai menyelam jauh ke dalam [Praktik Terbaik untuk Amazon MQ](#) dan [API REST Amazon MQ](#), dan kemudian [berencana untuk bermigrasi ke Amazon MQ](#).

## 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 instans broker kelas (m5, t3) dan ukuran (large, micro) adalah tipe instans broker (misalnya, mq.m5.large).

Topik

- [Langkah 1: membuat broker RabbitMQ](#)
- [Langkah 2: menghubungkan aplikasi berbasis JVM ke broker Anda](#)
- [Langkah 3: \(Opsional\) terhubung ke suatu AWS Lambda fungsi](#)
- [Langkah 4: hapus broker Anda](#)
- [Langkah selanjutnya](#)

## Langkah 1: membuat broker RabbitMQ


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

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 [Broker instans tunggal](#).
  - Deployment klaster RabbitMQ untuk ketersediaan tinggi adalah pengelompokan logis dari tiga node broker RabbitMQ di balik Penyeimbang Beban Jaringan, masing-masing membagikan pengguna, antrian, dan status terdistribusi di beberapa Availability Zone (AZ). Untuk informasi selengkapnya, lihat [Deployment klaster untuk ketersediaan tinggi](#).
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 Tipe instans broker (misalnya, mq.m5.large). Untuk informasi selengkapnya, lihat [Broker instance types](#).

 Note

Bagian Pengaturan tambahan menyediakan opsi untuk mengaktifkan CloudWatch log dan mengonfigurasi akses jaringan untuk broker Anda. Jika membuat broker RabbitMQ privat tanpa aksesibilitas publik, Anda harus memilih Virtual Private Cloud (VPC) dan mengonfigurasi grup keamanan untuk mengakses broker.

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.

6. Pilih Selanjutnya.
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.

	Name ▼	Status ▼	Deployment mode ▼	Instance type ▼
<input type="radio"/>	MyBroker	Running	Single-instance broker	mq.m5.large

## 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.amazonaws.com
```

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.amazonaws.com:5671
```

## Langkah 2: menghubungkan 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 developer](#).

### 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 RabbitMQ Java, lihat [Panduan API Klien RabbitMQ Java](#).

## Langkah 3: (Opsional) terhubung ke suatu AWS Lambda fungsi


AWS Lambda dapat terhubung ke dan mengonsumsi 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:
- Konfigurasi 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](#) di Panduan Pengembang.
3. [Konfigurasi broker Anda sebagai sumber acara](#) untuk fungsi Lambda menggunakan. AWS Management Console 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 peristiwa 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](#)

## Langkah 4: hapus broker Anda

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 AWS Management Console.

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.

## Langkah selanjutnya

Sekarang Anda telah membuat broker, menghubungkan aplikasi ke broker, serta mengirim dan menerima pesan, Anda mungkin ingin mencoba hal berikut:

- [Mengedit versi mesin broker, jenis instans, CloudWatchlog, dan preferensi pemeliharaan](#)
- [Listing brokers and viewing broker details](#)
- [Membuat dan mengelola pengguna broker ActiveMQ](#)
- [Rebooting a Broker](#)
- [Mengakses CloudWatch metrik untuk Amazon MQ](#)

Anda juga dapat mulai menyelami [Praktik Terbaik untuk Amazon MQ dan Amazon MQ REST API sebelum berencana untuk bermigrasi ke Amazon MQ](#).

# Mengelola broker Amazon MQ

Di bagian berikut, Anda dapat menemukan petunjuk untuk mengelola dan mempertahankan broker Amazon MQ.

## Topik

- [Mempertahankan broker Amazon MQ](#)
- [Meningkatkan versi mesin broker Amazon MQ](#)
- [Status broker](#)
- [Membuat daftar broker Amazon MQ dan melihat detail broker](#)
- [Mengakses konsol web broker tanpa aksesibilitas publik](#)
- [Melakukan boot ulang broker Amazon MQ](#)
- [Menghapus broker Amazon MQ](#)
- [Mengelola konfigurasi broker Amazon MQ](#)
- [Tipe instans](#)
- [Penandaan sumber daya](#)

## Mempertahankan broker Amazon MQ

Secara berkala, Amazon MQ melakukan pemeliharaan pada perangkat keras, sistem operasi, atau perangkat lunak mesin broker pesan. Durasi pemeliharaan bervariasi, tetapi dapat bertahan hingga dua jam, tergantung pada operasi yang dijadwalkan untuk broker pesan Anda. Misalnya, jika Anda telah mengaktifkan [upgrade versi mesin minor otomatis](#), atau mengubah jenis instans broker, Amazon MQ akan menerapkan perubahan Anda selama jendela pemeliharaan terjadwal berikutnya.

Untuk meminimalkan waktu henti selama jendela pemeliharaan, sebaiknya pilih mode penyebaran broker dengan ketersediaan tinggi di beberapa Availability Zone (AZ). Bergantung pada jenis mesin broker Anda, Amazon MQ menyediakan mode penyebaran Multi-AZ berikut.

- Amazon MQ untuk ActiveMQ - Amazon MQ [untuk ActiveMQ menyediakan penerapan aktif/siaga untuk ketersediaan tinggi](#). Dalam mode aktif/siaga, Amazon MQ melakukan operasi pemeliharaan satu instans pada satu waktu, memastikan bahwa setidaknya satu instance tetap tersedia. Selain itu, Anda dapat mengonfigurasi [jaringan broker](#) dengan jendela pemeliharaan yang tersebar sepanjang minggu.

- Amazon MQ untuk RabbitMQ — [Amazon MQ untuk RabbitMQ menyediakan penerapan cluster untuk ketersediaan tinggi](#). Dalam penerapan cluster, Amazon MQ melakukan operasi pemeliharaan, satu node pada satu waktu, menjaga setidaknya dua node yang berjalan setiap saat.

Untuk informasi selengkapnya tentang Amazon MQ merekomendasikan praktik terbaik untuk memastikan broker Anda bekerja secara efektif selama, dan setelah jendela pemeliharaan, lihat dokumentasi berikut untuk jenis mesin broker Anda.

- [the section called “Praktik terbaik Amazon MQ for ActiveMQ”](#)
- [the section called “Amazon MQ untuk praktik terbaik RabbitMQ”](#)

Anda dapat menjadwalkan pemeliharaan terjadi seminggu sekali pada waktu tertentu yang berlangsung hingga dua jam. Ini menetapkan jendela untuk tindakan pemeliharaan dari Amazon MQ untuk dijadwalkan dan dimulai.

Anda dapat menjadwalkan jendela pemeliharaan saat pertama kali membuat broker Anda, atau dengan memperbarui preferensi broker Anda. Topik berikut menjelaskan penyesuaian jendela pemeliharaan broker menggunakan AWS Management Console, AWS CLI, dan Amazon MQ API.

Topik

- [Menyesuaikan jendela pemeliharaan broker](#)

## Menyesuaikan jendela pemeliharaan broker

Selama waktu jendela pemeliharaan yang Anda pilih, Amazon MQ akan melakukan perubahan yang tertunda, seperti peningkatan versi minor otomatis. Untuk menyesuaikan jendela pemeliharaan broker, Anda dapat menggunakan AWS Management Console, AWS CLI, atau Amazon MQ API.

### Important

Anda hanya dapat menyesuaikan jendela pemeliharaan broker hingga empat kali sebelum jendela pemeliharaan terjadwal berikutnya. Amazon MQ menerapkan batas empat penyesuaian jendela pemeliharaan untuk memastikan bahwa patch perangkat lunak dan keamanan penting, serta peningkatan perangkat keras yang penting, tidak ditangguhkan dan ditunda tanpa batas waktu.

Setelah jendela pemeliharaan broker selesai, Amazon MQ mengatur ulang batas, memungkinkan Anda untuk menyesuaikan jadwal sebelum jendela pemeliharaan berikutnya terjadi.

Ketersediaan broker tidak terpengaruh saat menyesuaikan jendela pemeliharaan broker.

## AWS Management Console

Untuk menyesuaikan jendela pemeliharaan broker dengan menggunakan AWS Management Console

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. Gulir ke bagian bawah halaman, dan pilih Simpan. Jendela pemeliharaan segera disesuaikan.
6. Pada halaman detail broker, di bawah jendela Pemeliharaan, verifikasi bahwa jadwal pilihan baru Anda ditampilkan.

## 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-3i45j6k17819, ID broker akan menjadi b-1234a5b6-78cd-901e-2fgh-3i45j6k17819.

- `--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
```

## API Amazon MQ

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 [WeeklyStartTimes](#) sumber daya dalam payload permintaan.

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

- (Opsional) Gunakan operasi [DescribeBroker](#) API 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
```

## Meningkatkan versi mesin broker Amazon MQ

Amazon MQ menyediakan versi mesin broker baru untuk semua jenis mesin broker yang didukung. Versi mesin baru mungkin menyertakan patch keamanan, perbaikan bug, dan peningkatan mesin broker lainnya. Ketika Amazon MQ mendukung versi mesin baru, Anda dapat mengontrol cara dan waktu untuk meningkatkan broker.

Versi mesin broker diatur sebagai X.Y.Z. Dalam implementasi Amazon MQ dari setiap jenis mesin, X.Y dianggap sebagai versi utama dan Z dianggap sebagai versi minor. Ada dua jenis peningkatan:

- Peningkatan versi utama — Terjadi ketika nomor versi mesin utama berubah. Contohnya, meningkatkan dari versi 1.0 ke versi 1.1 dianggap sebagai peningkatan versi utama.
- Peningkatan versi minor — Terjadi ketika nomor versi mesin minor hanya nomor versi mesin minor yang berubah. Contohnya, meningkatkan dari versi 1.1.0 ke versi 1.1.1 dianggap sebagai peningkatan versi minor.

Untuk informasi selengkapnya tentang manajemen versi utama dan minor untuk setiap jenis mesin broker tertentu, lihat topik berikut.

- [the section called “Manajemen versi”](#)

- [the section called “Manajemen versi”](#)

Saat Anda mengaktifkan [peningkatan versi minor otomatis](#), Amazon MQ meningkatkan broker Anda ke versi minor baru saat versi tersebut tersedia. Peningkatan versi minor otomatis terjadi hanya jika broker menjalankan versi mesin minor yang lebih rendah dari versi minor baru yang direkomendasikan. Untuk peningkatan utama, Anda harus meningkatkan versi mesin secara manual.

Peningkatan versi manual dan otomatis terjadi selama jendela pemeliharaan terjadwal atau setelah Anda [melakukan boot ulang broker](#).

Topik berikut menjelaskan cara meningkatkan versi mesin broker secara manual, dan mengaktifkan peningkatan versi minor otomatis.

Topik

- [Meningkatkan versi mesin secara manual](#)
- [Otomatis meningkatkan versi mesin minor](#)

## Meningkatkan versi mesin secara manual


Untuk secara manual meningkatkan versi mesin broker ke versi utama atau minor yang baru, Anda dapat menggunakan AWS Management Console, AWS CLI, atau API Amazon MQ.

### AWS Management Console

Untuk meningkatkan versi mesin broker menggunakan AWS Management Console

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 Versi mesin broker, pilih nomor versi baru dari daftar dropdown.
5. Gulir ke bagian bawah halaman, lalu pilih Jadwalkan perubahan.
6. 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

Broker Anda akan menjadi offline saat sedang di-boot ulang.

7. Pilih Terapkan untuk menyelesaikan penerapan perubahan.

## AWS CLI

Untuk meningkatkan versi mesin 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-3i45j6k17819`, ID broker akan menjadi `b-1234a5b6-78cd-901e-2fgh-3i45j6k17819`.
  - `--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 melakukan boot ulang broker jika Anda ingin segera meningkatkan 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

Broker Anda akan menjadi offline saat sedang di-boot ulang.

## 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 dalam 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 [RebootBroker](#) API untuk mem-boot ulang broker, jika Anda ingin segera meningkatkan 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

Broker Anda akan menjadi offline saat sedang di-boot ulang.

## Otomatis meningkatkan versi mesin minor

Anda dapat mengontrol apakah peningkatan versi minor otomatis diaktifkan untuk broker ketika Anda pertama kali membuat broker, atau dengan mengubah preferensi broker. Untuk mengaktifkan peningkatan versi minor otomatis bagi broker yang sudah ada, Anda dapat menggunakan AWS Management Console, AWS CLI, atau API Amazon MQ.

### AWS Management Console

Untuk mengaktifkan peningkatan versi minor otomatis menggunakan AWS Management Console

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 Pemeliharaan, pilih Aktifkan peningkatan versi minor otomatis.

#### Note

Jika opsi sudah dipilih, Anda tidak perlu melakukan perubahan apa pun.

5. Pilih Simpan di bagian bawah halaman.

### AWS CLI

Untuk mengaktifkan peningkatan versi minor otomatis menggunakan perintah CLI [update-broker](#) dan tentukan parameter berikut. AWS CLI

- `--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`.
- `--auto-minor-version-upgrade` – Mengaktifkan opsi peningkatan versi minor otomatis.

```
aws mq update-broker --broker-id broker-id --auto-minor-version-upgrade
```

Jika Anda ingin menonaktifkan peningkatan versi minor otomatis untuk broker, gunakan parameter `--no-auto-minor-version-upgrade`.

## API Amazon MQ

Untuk mengaktifkan peningkatan versi minor otomatis Amazon MQ minor otomatis menggunakan [UpdateBroker](#) API. 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 dalam 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 properti `autoMinorVersionUpgrade` di muatan permintaan untuk mengaktifkan peningkatan versi minor otomatis.

```
{
  "autoMinorVersionUpgrade": "true"
}
```

Jika Anda ingin menonaktifkan peningkatan versi minor otomatis untuk broker, atur `"autoMinorVersionUpgrade": "false"` dalam muatan permintaan.

## Status broker

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.

Konsol	API	Deskripsi
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 di <a href="#">the section called “Pemecahan masalah: Kode yang diperlukan tindakan Amazon MQ”</a> .

## Membuat daftar broker Amazon MQ dan melihat detail broker

Ketika Anda meminta Amazon MQ membuat broker, proses pembuatan dapat berlangsung sekitar 15 menit.

Contoh berikut menunjukkan cara mengonfirmasi keberadaan broker Anda dengan membuat daftar broker di wilayah saat ini menggunakan AWS Management Console.

### Untuk membuat daftar broker dan melihat detail broker

1. Masuk ke [konsol Amazon MQ](#).

Broker Anda di wilayah saat ini ada dalam daftar.



Brokers (3) <a href="#">Info</a>							
<input type="text" value="Search name"/> <span style="float: right;">&lt; 1 &gt; </span>							
	Name ▲	Creation time (Local) ▼	Status ▼	Broker engine ▼	Deployment mode ▼	Instance type ▼	
<input type="radio"/>	<a href="#">MyBroker</a>	Oct 27, 2020 9:39 AM	Running	ActiveMQ	Active/standby broker	mq.m5.large	
<input type="radio"/>	<a href="#">MyBroker2</a>	Oct 27, 2020 9:40 AM	Running	RabbitMQ	Single-instance broker	mq.m5.large	
<input type="radio"/>	<a href="#">MyBroker3</a>	Oct 27, 2020 9:38 AM	Running	RabbitMQ	Cluster deployment	mq.m5.large	

Informasi berikut ini ditampilkan untuk masing-masing broker:

- Nama
- Tanggal pembuatan
- [Status](#)
- Modus penyebaran
- [Tipe instans](#)

## 2. Pilih nama broker Anda.


Untuk broker ActiveMQ, di **MyBroker** halaman, [dikonfigurasi](#) Detail ditampilkan untuk broker Anda:

Details			
ARN <a href="#">Info</a> arn:aws:mq:us-west-2:123878009876:broker:MyBroker:b-2f91ed40-de60-40b2-9141-ddce16cb0a0f			
<b>Specifications</b>  Broker status <b>Running</b>  Broker name <b>MyBroker</b>  Broker instance type <a href="#">Info</a> <b>mq.m5.large</b>  Deployment mode <a href="#">Info</a> <b>Active/standby broker</b>  Storage type <a href="#">Info</a> <b>Amazon Elastic File System</b>  Broker engine <a href="#">Info</a> <b>ActiveMQ</b>  Broker engine version <b>5.15.12</b>	<b>Configuration</b>  Configuration name <b>MyBroker-configuration</b>  Configuration revision <a href="#">Revision 1 - Auto-generated default for MyBroker-configuration on ActiveMQ 5.15.12</a>  <b>CloudWatch Logs</b>  General <b>Disabled - <a href="#">Logs</a></b>  Audit <b>Disabled - <a href="#">Logs</a></b>	<b>Security and network</b>  VPC <a href="#">Info</a> <b>vpc-286cba5b <a href="#">↗</a></b>  Subnet(s) <a href="#">Info</a> <b>subnet-4388bb98 <a href="#">↗</a></b> <b>subnet-7942b82g <a href="#">↗</a></b>  Security group(s) <a href="#">Info</a> <b>sg-1abc5867 <a href="#">↗</a></b>  Public accessibility <a href="#">Info</a> <b>Yes</b>  IP Addresses <b>53.208.204.167</b> <b>46.290.203.267</b>	<b>Maintenance</b>  Automatic minor version upgrade <b>Yes</b>  Maintenance window <b>Saturday 19:00 - 21:00 UTC</b>

Untuk Amazon MQ untuk broker RabbitMQ, Anda dapat melihat pengaturan yang Anda pilih di **MyBroker2** halaman, di bawah Detail bagian seperti yang ditunjukkan pada berikut ini.

## Details

### ARN [Info](#)

 `arn:aws:mq:us-west-2:123413139898:broker:MyBroker2:b-751396a6-e097-4e7f-85e4-de98a5598869`

### Broker name

MyBroker2

### Broker status

Running

### Creation time

Oct 27, 2020 9:40 AM

### Broker engine [Info](#)

RabbitMQ

### Deployment mode [Info](#)

Single-instance broker

### Broker instance type [Info](#)

mq.m5.large

### Broker engine version

3.8.6

### CloudWatch Logs

Disabled - [Logs](#)

### Maintenance

#### Automatic minor version upgrade

Yes

#### Maintenance window

Tuesday 18:00 - 20:00 UTC

### Security and network

#### VPC [Info](#)

vpc-111cca5b [↗](#)

#### Subnet(s) [Info](#)

subnet-8vr11jn8 [↗](#)

#### Public accessibility [Info](#)

Yes

Dalam bagian Detail, informasi berikut ini ditampilkan:

- Di bagian Koneksi, untuk broker Amazon MQ for ActiveMQ, URL konsol web dan titik akhir protokol tingkat kawat.

### Connections

Access your queues and topics and connect your application to the broker. If you disable public accessibility for your broker, your endpoints are reachable only within a VPC.



#### Enable connections to your broker

To be able to access your broker's ActiveMQ Web Console URL or wire-level protocol endpoints, you must configure one of your security groups to allow inbound traffic. [Detailed instructions](#)



### ActiveMQ Web Console

In an active/standby deployment, only one of the ActiveMQ Web Console URLs is active at a time.

<https://b-2f91ed40-de60-40b2-9141-ddce16cb0a0f-1.mq.us-west-2.amazonaws.com:8162>

<https://b-2f91ed40-de60-40b2-9141-ddce16cb0a0f-2.mq.us-west-2.amazonaws.com:8162>

### Endpoints

In an active/standby deployment, only one of the endpoints in each pair is active at a time. You can allow your application to establish connection to either endpoint by using the ActiveMQ Failover Transport.

OpenWire	<code>ssl://b-2f91ed40-de60-40b2-9141-ddce16cb0a0f-1.mq.us-west-2.amazonaws.com:61617</code>	<a href="#">Copy failover string (Java)</a>
	<code>ssl://b-2f91ed40-de60-40b2-9141-ddce16cb0a0f-2.mq.us-west-2.amazonaws.com:61617</code>	
AMQP	<code>amqp+ssl://b-2f91ed40-de60-40b2-9141-ddce16cb0a0f-1.mq.us-west-2.amazonaws.com:5671</code>	<a href="#">Copy failover string (Java)</a>
	<code>amqp+ssl://b-2f91ed40-de60-40b2-9141-ddce16cb0a0f-2.mq.us-west-2.amazonaws.com:5671</code>	
STOMP	<code>stomp+ssl://b-2f91ed40-de60-40b2-9141-ddce16cb0a0f-1.mq.us-west-2.amazonaws.com:61614</code>	<a href="#">Copy failover string (Java)</a>
	<code>stomp+ssl://b-2f91ed40-de60-40b2-9141-ddce16cb0a0f-2.mq.us-west-2.amazonaws.com:61614</code>	
MQTT	<code>mqtt+ssl://b-2f91ed40-de60-40b2-9141-ddce16cb0a0f-1.mq.us-west-2.amazonaws.com:8883</code>	<a href="#">Copy failover string (Java)</a>
	<code>mqtt+ssl://b-2f91ed40-de60-40b2-9141-ddce16cb0a0f-2.mq.us-west-2.amazonaws.com:8883</code>	
WSS	<code>wss://b-2f91ed40-de60-40b2-9141-ddce16cb0a0f-1.mq.us-west-2.amazonaws.com:61619</code>	<a href="#">Copy failover string (Java)</a>
	<code>wss://b-2f91ed40-de60-40b2-9141-ddce16cb0a0f-2.mq.us-west-2.amazonaws.com:61619</code>	

Di bagian Koneksi, untuk broker Amazon MQ for RabbitMQ, URL konsol web dan titik akhir AMQP aman.

### Connections

Access your queues and exchanges and connect your application to the broker. If you disable public accessibility for your broker, your endpoints are reachable only within a VPC.

#### RabbitMQ web console

<https://b-c8349341-ec91-4a78-ad9c-a57f23f235bb.mq.us-west-2.amazonaws.com>

Endpoints	
Name	URL
AMQP	<a href="https://b-c8349341-ec91-4a78-ad9c-a57f23f235bb.mq.us-west-2.amazonaws.com:5671">amqps://b-c8349341-ec91-4a78-ad9c-a57f23f235bb.mq.us-west-2.amazonaws.com:5671</a>

- Untuk broker Amazon MQ for ActiveMQ, di bagian Pengguna, [pengguna](#) yang dikaitkan dengan broker

#### Important

Mengelola pengguna melalui AWS Management Console dan API Amazon MQ tidak didukung untuk broker Amazon MQ for RabbitMQ.

## Mengakses konsol web broker tanpa aksesibilitas publik

Jika Anda menonaktifkan aksesibilitas publik untuk broker, Anda harus melakukan langkah-langkah berikut agar dapat mengakses konsol web broker.

#### Note

Nama-nama VPC dan grup keamanan khusus untuk contoh berikut.

## Prasyarat

Untuk melakukan langkah-langkah berikut, Anda harus mengonfigurasi hal berikut:

- VPC
  - VPC tanpa gateway internet, yang broker Amazon MQ dilampirkan, bernama `private-vpc`.
  - VPC kedua, dengan gateway internet, bernama `public-vpc`.
  - Kedua VPC harus terhubung (misalnya, menggunakan [Peering VPC](#)) sehingga instans Amazon EC2 di VPC publik dapat berkomunikasi dengan instans EC2 di VPC privat.

- Jika Anda menggunakan peering VPC, tabel rute untuk kedua VPC harus dikonfigurasi untuk koneksi peering.
- Grup Keamanan
  - Grup keamanan yang digunakan untuk membuat broker Amazon MQ, bernama `private-sg`.
  - Grup keamanan kedua yang digunakan untuk instans EC2 di VPC `public-vpc`, bernama `public-sg`.
  - `private-sg` harus mengizinkan koneksi masuk dari `public-sg`. Kami merekomendasikan untuk membatasi grup keamanan ini ke port 8162 untuk ActiveMQ, dan port 443 untuk RabbitMQ.
  - `public-sg` harus mengizinkan koneksi masuk dari mesin Anda di port 22.

## Untuk mengakses konsol web broker pada Broker tanpa Aksesibilitas Publik

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 [FoxyProxy](#) pada mesin Anda.
5. Konfigurasi klien proxy 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.

# 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.

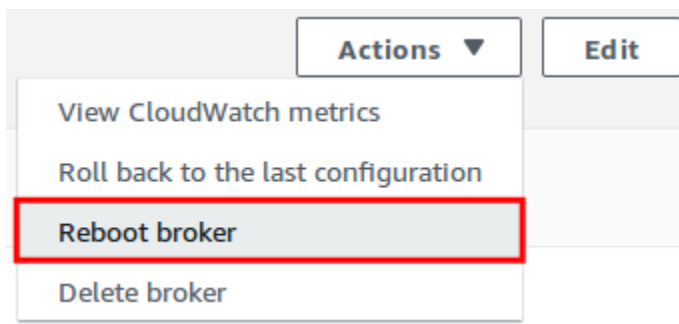
Contoh berikut menunjukkan cara melakukan boot ulang broker Amazon MQ menggunakan AWS Management Console.

## 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 Tindakan, 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 akan menggunakannya dalam waktu dekat), praktik terbaiknya adalah menghapus broker dari Amazon MQ untuk mengurangi biaya AWS.

Contoh berikut menunjukkan cara menghapus broker menggunakan AWS Management Console.

### 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 memilih Hapus.

Menghapus broker berlangsung sekitar 5 menit.

## Mengelola konfigurasi broker Amazon MQ

SEBUAH konfigurasi berisi semua pengaturan untuk broker Anda. Anda dapat membuat konfigurasi sebelum membuat broker. Anda kemudian dapat menerapkan konfigurasi ke satu atau lebih broker

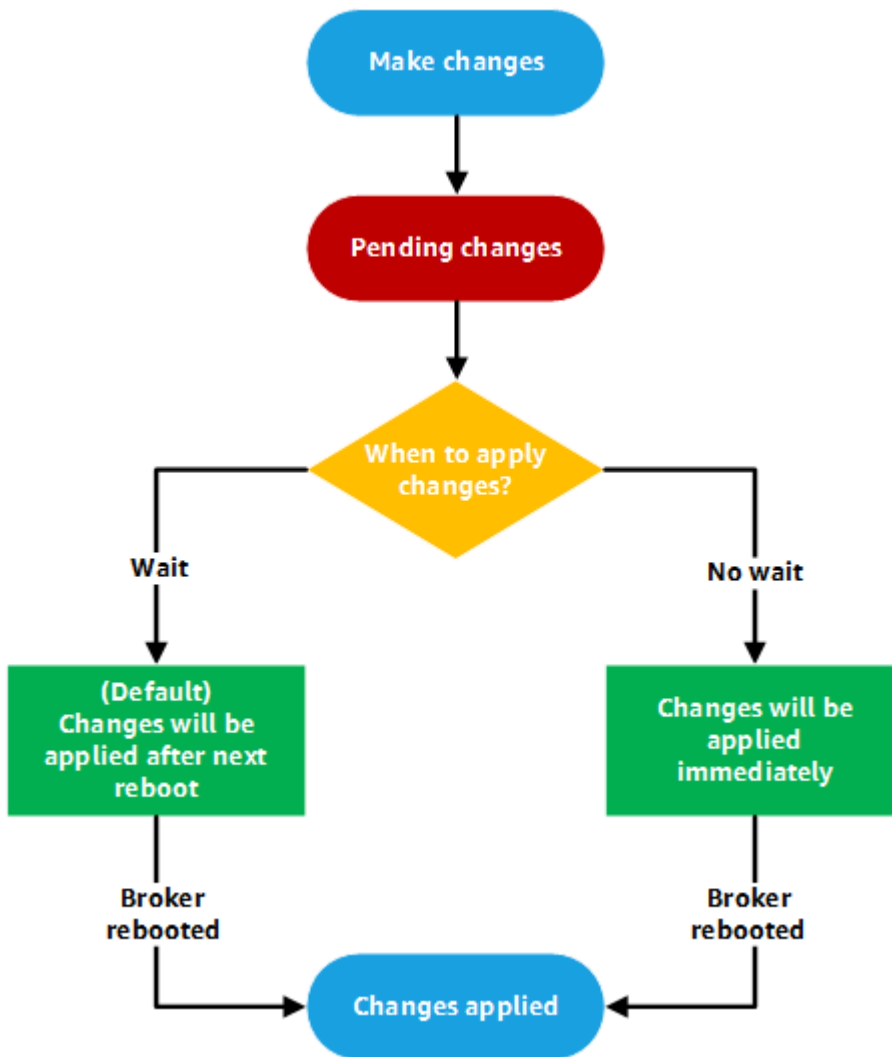
### Siklus hidup konfigurasi broker Amazon MQ

Pembuatan perubahan pada revisi konfigurasi atau pengguna ActiveMQ tidak akan segera menerapkan perubahan. Untuk menerapkan perubahan Anda, Anda harus menunggu jendela pemeliharaan berikutnya atau [reboot broker](#). Untuk informasi selengkapnya, lihat [Siklus hidup konfigurasi broker Amazon MQ](#).

Diagram berikut menggambarkan siklus hidup konfigurasi.

#### Important

Jendela pemeliharaan terjadwal berikutnya memicu boot ulang. Jika broker di-boot ulang sebelum jendela pemeliharaan terjadwal berikutnya, perubahan diterapkan setelah boot ulang.



Untuk `ActiveMQ`, konfigurasi berisi semua pengaturan untuk broker Anda dalam format XHTML (mirip dengan `ActiveMQactivemq.xml` berkas). Untuk informasi selengkapnya tentang membuat, menerapkan, dan mengedit konfigurasi broker `ActiveMQ`, lihat [Creating and applying broker configurations](#).

Untuk `RabbitMQ`, konfigurasi mengatur semua pengaturan untuk broker Anda dalam format Sotong. Untuk informasi lebih lanjut tentang membuat, menerapkan, dan mengedit konfigurasi broker `RabbitMQ`, lihat [Creating and applying broker configurations](#).

## Tipe instans

Deskripsi gabungan dari instans broker kelas (`m5`, `t3`) dan ukuran (`large`, `micro`) adalah tipe instans broker (misalnya, `mq.m5.large`). Tabel berikut mencantumkan tipe instans broker Amazon MQ yang tersedia untuk setiap jenis mesin yang didukung.


## Topik

- [Tipe instans Amazon MQ for ActiveMQ](#)
- [Tipe instans Amazon MQ for RabbitMQ](#)

## Tipe instans Amazon MQ for ActiveMQ

### Important

Anda dapat menggunakan Amazon EBS hanya dengan keluarga tipe instans broker mq.m5. Untuk informasi selengkapnya, lihat [Storage](#).

Tipe Instans	vCPU	Memori (GiB)	Kinerja Jaringan	Catatan
mq.t2.micro	1	1	Rendah	<p>Menggunakan tipe instans mq.t2.micro untuk evaluasi dasar Amazon MQ. Tipe instans tunggal saja) memenuhi syarat untuk <a href="#">AWS Tingkat Gratis</a>.</p> <div data-bbox="1258 1365 1510 1879" style="border: 1px solid #add8e6; border-radius: 10px; padding: 10px; margin-top: 10px;"> <p> Note</p> <p>Penggunaan tipe instans mq.t2.micro tunduk pada <a href="#">kredit CPU dan</a></p> </div>




Tipe Instans	vCPU	Memori (GiB)	Kinerja Jaringan	Catatan
				<p><a href="#">performa dasar</a>— dengan kemampuan untuk burst di atas tingkat dasar (untuk informasi selengkapnya, lihat metrik <a href="#">CpuCreditBalance</a> ).</p> <p>Jika aplikasi Anda membutuhkan performan tetap, pertimbangkan untuk menggunakan tipe instans <code>mq.m5.large</code> .</p>

Tipe Instans	vCPU	Memori (GiB)	Kinerja Jaringan	Catatan
mq.t3.micro	2	1	Rendah	Menggunakan tipe instans mq.t3.micro untuk evaluasi dasar Amazon MQ. Tipe instans ini (broker instans tunggal saja) memenuhi syarat untuk <a href="#">Tingkat Gratis AWS</a> .
mq.m4.large	2	8	Sedang	Menggunakan tipe instans mq.m4.large untuk kompatibilitas dengan deployment broker yang ada. Kami merekomendasikan penggunaan instans mq.m5.* untuk broker baru.

Tipe Instans	vCPU	Memori (GiB)	Kinerja Jaringan	Catatan
mq.m5.large	2	8	Tinggi	Menggunakan instans mq.m5.large untuk pengembangan reguler, pengujian, dan beban kerja produksi.

Tipe Instans	vCPU	Memori (GiB)	Kinerja Jaringan	Catatan
mq.m5.xlarge	4	16	Tinggi	Menggunakan tipe instans mq.m5.xlarge, mq.m5.2xlarge, dan mq.m5.4xlarge untuk pengembangan reguler, pengujian, dan beban kerja produksi yang memerlukan throughput tinggi.
mq.m5.2xlarge	8	32	Tinggi	
mq.m5.4xlarge	16	64	Tinggi	

 **Note**

Ketika sistem Anda menggunakan pesan persisten, throughput tergantung pada seberapa cepat pesan dikonsumsi.

Tipe Instans	vCPU	Memori (GiB)	Kinerja Jaringan	Catatan
				<p>Jika pesan tidak segera dikonsumsi, penggunaan tipe instans yang lebih besar dengan pesan persisten mungkin tidak meningkatkan throughput sistem. Dalam kasus ini, kami merekomendasikan mengatur atribut <code>concurrentStoreAndDispatchQueues</code> ke <code>false</code>. Untuk informasi</p>

Tipe Instans	vCPU	Memori (GiB)	Kinerja Jaringan	Catatan
				selengkapnya, lihat <a href="#">Menonaktifkan Penyimpanan dan Pengiriman Bersamaan untuk Antrean dengan Konsumen Lambat.</a>


Untuk informasi selengkapnya tentang pertimbangan throughput, lihat [Memilih Tipe Instans Broker yang Tepat untuk Throughput Terbaik.](#)

## Tipe instans Amazon MQ for RabbitMQ

### Important

Anda tidak dapat menurunkan peringkat broker dari `mq.m5` tipe instans ke `mq.t3.micro` tipe instans.

Tipe Instans	vCPU	Memori (GiB)	Kinerja Jaringan	Catatan
<code>mq.t3.micro</code>	2	1	Rendah	Menggunakan tipe instans <code>mq.t3.micro</code> untuk evaluasi dasar Amazon MQ. Tipe instans

Tipe Instans	vCPU	Memori (GiB)	Kinerja Jaringan	Catatan
				<p>ini (broker instans tunggal saja) memenuhi syarat untuk <a href="#">Tingkat Gratis AWS</a>.</p> <div data-bbox="1258 525 1510 1029" style="border: 1px solid #f08080; border-radius: 10px; padding: 10px; background-color: #fff9e6;"> <p> Important Parameter tipe instans mq.t3.micro tidak mendukung <a href="#">deployment kluster</a>.</p> </div>
mq.m5.large	2	8	Tinggi	Menggunakan instans mq.m5.large untuk pengembangan reguler, pengujian, dan beban kerja produksi.

Tipe Instans	vCPU	Memori (GiB)	Kinerja Jaringan	Catatan
mq.m5.xlarge	4	16	Tinggi	Menggunakan tipe instans mq.m5.xlarge
mq.m5.2xlarge	8	32	Tinggi	mq.m5.xlarge , mq.m5.2xlarge , dan
mq.m5.4xlarge	16	64	Tinggi	mq.m5.4xlarge untuk pengembangan reguler, pengujian, dan beban kerja produksi yang memerlukan throughput tinggi.

## Penandaan sumber daya

Amazon MQ mendukung penandaan sumber daya untuk membantu melacak alokasi biaya Anda. Anda dapat menandai sumber daya saat membuatnya, atau dengan melihat detail sumber daya tersebut.

### Topik

- [Penandaan untuk Alokasi Biaya](#)
- [Mengelola Tanda di Konsol Amazon MQ](#)
- [Mengelola Menggunakan Tindakan API Amazon MQ](#)

## Penandaan untuk Alokasi Biaya

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 tanda alokasi biaya untuk



mengelola tagihan AWS guna merefleksikan struktur biaya. Untuk melakukannya, daftar untuk membuat tagihan akun AWS Anda menyertakan kunci dan nilai tanda. 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.

## Mengelola Tanda di Konsol Amazon MQ

### Menambahkan Tanda ke Sumber Daya Baru

Amazon MQ memungkinkan Anda menambahkan tanda ke sumber daya saat dibuat. Anda dapat menambahkan tanda dengan cepat ke sumber daya yang dibuat di konsol Amazon MQ.

Untuk menambahkan tanda saat Anda membuat broker baru:

1. Dari halaman Buat broker, pilih Pengaturan tambahan.
2. Di bawah Tanda, pilih Tambah tanda.
3. Masukkan pasangan Kunci dan Nilai.

## Tags - optional

You can add tags to describe your broker. A tag consists of a case-sensitive key-value pair. [Learn more](#) 

Key

Value - optional

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.

## Melihat dan Mengelola Tanda untuk Sumber Daya yang Ada

Amazon MQ memungkinkan Anda melihat dan mengelola tanda untuk sumber daya di konsol Amazon MQ. Anda dapat mengelola tanda untuk sumber daya individu dengan mengedit tanda pada halaman detail sumber daya tersebut. Untuk mengedit tanda pada sumber Amazon MQ:

1. Pilih Broker atau Konfigurasi di konsol Amazon MQ.

Di bawah bagian Tanda, tinjau tanda yang ada untuk sumber daya tersebut.

2. Untuk menambahkan tanda baru atau mengelola tanda yang ada, pilih Edit (atau Buat tanda jika tidak ada tanda).
3. Memperbarui tanda untuk sumber daya Anda:
  - Untuk memodifikasi tanda yang ada, edit Kunci dan Nilai.
  - Untuk menghapus tanda yang ada, pilih Hapus.
  - Untuk menambahkan tanda baru, pilih Tambah tanda lalu masukkan Kunci dan Nilai.

#### 4. Pilih Simpan.

## Mengelola Menggunakan Tindakan API Amazon MQ

Amazon MQ memungkinkan Anda melihat dan mengelola tanda sumber daya menggunakan REST API.

Untuk informasi selengkapnya, lihat [Referensi REST API Amazon MQ](#).

# Bekerja dengan 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 AWS Management Console, REST API Amazon MQ, 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 REST API Amazon MQ, lihat [Referensi REST API Amazon MQ](#).

Topik

- [Mesin ActiveMQ](#)
- [Tutorial ActiveMQ](#)
- [Praktik terbaik Amazon MQ for ActiveMQ](#)
- [Replikasi data Lintas Wilayah untuk Amazon MQ untuk ActiveMQ](#)
- [Kuota di Amazon MQ untuk ActiveMQ](#)

## Mesin ActiveMQ

Bagian ini menjelaskan elemen dasar dari broker ActiveMQ, memberikan gambaran umum tentang opsi arsitektur broker ActiveMQ, menjelaskan parameter konfigurasi broker, dan menawarkan contoh kerja menggunakan Java Message Service (JMS).

Topik

- [Elemen dasar](#)
- [Arsitektur broker](#)
- [Amazon MQ untuk konfigurasi broker ActiveMQ](#)
- [Mengelola versi mesin Amazon MQ for ActiveMQ](#)
- [Contoh kerja menggunakan Java Message Service \(JMS\) dengan ActiveMQ](#)

## Elemen dasar

Bagian ini memperkenalkan konsep-konsep kunci penting untuk memahami ActiveMQ di Amazon MQ.

Topik

- [Pialang](#)
- [Jenis instans broker](#)
- [Konfigurasi](#)
- [Pengguna](#)
- [Penyimpanan](#)

### Pialang

Broker adalah lingkungan broker pesan yang berjalan di Amazon MQ. Ini adalah blok bangunan dasar Amazon MQ. Deskripsi gabungan dari instans broker kelas (m5, t3) dan ukuran (large, micro) adalah tipe instans broker (misalnya, mq.m5.large). Untuk informasi selengkapnya, lihat [Broker instance types](#).

- SEBUAH broker satu contoh 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 [Broker Architecture](#).

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:

- [Creating and configuring a broker](#)
- [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-INJAK](#)
- STOMP berakhir [WebSocket](#)

Atribut

Broker ActiveMQ memiliki beberapa atribut, misalnya:

- Nama (MyBroker)
- ID (b-1234a5b6-78cd-901e-2fgh-3i45j6k17819)
- Amazon Resource Name (ARN) (arn:aws:mq:us-east-2:123456789012:broker:MyBroker:b-1234a5b6-78cd-901e-2fgh-3i45j6k17819)
- URL Konsol Web ActiveMQ (https://b-1234a5b6-78cd-901e-2fgh-3i45j6k17819-1.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-3i45j6k17819-1.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 broker aktif/siaga, Amazon MQ menyediakan dua URL Konsol Web ActiveMQ, tetapi hanya satu URL 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](#)

## Jenis instans broker

### Important

Anda dapat menggunakan Amazon EBS hanya dengan keluarga tipe instans broker mq.m5. Untuk informasi selengkapnya, lihat [Storage](#).

Tipe Instans	vCPU	Memori (GiB)	Kinerja Jaringan	Catatan
mq.t2.micro	1	1	Rendah	<p>Menggunakan tipe instans mq.t2.micro untuk evaluasi dasar Amazon MQ. Jenis instans ini (hanya broker instans tunggal) memenuhi syarat untuk <a href="#">AWS Tingkat Gratis</a>.</p> <div data-bbox="1258 1222 1511 1885" style="border: 1px solid #add8e6; border-radius: 10px; padding: 10px; margin-top: 10px;"> <p> Note</p> <p>Penggunaan tipe instans mq.t2.micro tunduk pada <a href="#">kredit CPU dan performa dasar</a>—dengan</p> </div>




Tipe Instans	vCPU	Memori (GiB)	Kinerja Jaringan	Catatan
				<p>kemampuan untuk burst di atas tingkat dasar (untuk informasi selengkapnya, lihat metrik <a href="#">CpuCreditBalance</a> ).</p> <p>Jika aplikasi Anda membutuhkan an performa tetap, pertimbangkan untuk menggunakan an tipe instans <code>mq.m5.large</code> .</p>

Tipe Instans	vCPU	Memori (GiB)	Kinerja Jaringan	Catatan
mq.t3.micro	2	1	Rendah	Menggunakan tipe instans mq.t3.micro untuk evaluasi dasar Amazon MQ. Tipe instans ini (broker instans tunggal saja) memenuhi syarat untuk <a href="#">Tingkat Gratis AWS</a> .
mq.m4.large	2	8	Sedang	Menggunakan tipe instans mq.m4.large untuk kompatibilitas dengan deployment broker yang ada. Kami merekomendasikan penggunaan instans mq.m5.* untuk broker baru.

Tipe Instans	vCPU	Memori (GiB)	Kinerja Jaringan	Catatan
mq.m5.large	2	8	Tinggi	Menggunakan instans mq.m5.large untuk pengembangan reguler, pengujian, dan beban kerja produksi.

Tipe Instans	vCPU	Memori (GiB)	Kinerja Jaringan	Catatan
mq.m5.xlarge	4	16	Tinggi	Menggunakan tipe instans mq.m5.xlarge, mq.m5.2xlarge, dan mq.m5.4xlarge untuk pengembangan reguler, pengujian, dan beban kerja produksi yang memerlukan throughput tinggi.
mq.m5.2xlarge	8	32	Tinggi	
mq.m5.4xlarge	16	64	Tinggi	

 **Note**

Ketika sistem Anda menggunakan an pesan persisten, throughput tergantung pada seberapa cepat pesan dikonsumsi.

Tipe Instans	vCPU	Memori (GiB)	Kinerja Jaringan	Catatan
				<p>Jika pesan tidak segera dikonsumsi, penggunaan tipe instans yang lebih besar dengan pesan persisten mungkin tidak meningkatkan throughput sistem. Dalam kasus ini, kami merekomendasikan mengatur atribut <code>concurrentStoreAndDispatchQueues</code> ke <code>false</code>. Untuk informasi</p>

Tipe Instans	vCPU	Memori (GiB)	Kinerja Jaringan	Catatan
				<p>selengkapnya, lihat <a href="#">Menonaktifkan Penyimpanan dan Pengiriman Bersamaan untuk Antrean dengan Konsumen Lambat.</a></p>

Untuk informasi selengkapnya tentang pertimbangan throughput, lihat [Memilih Tipe Instans Broker yang Tepat untuk Throughput Terbaik.](#)

## Konfigurasi

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.

### 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](#). Untuk informasi selengkapnya, lihat [Siklus hidup konfigurasi broker Amazon MQ](#).

Saat ini, Anda tidak dapat menghapus konfigurasi.

Untuk informasi tentang membuat, mengedit, dan mengelola konfigurasi, lihat hal berikut:

- [Creating and applying broker configurations](#)
- [Konfigurasi](#)
- [Amazon MQ Broker Configuration Parameters](#)

Untuk melacak perubahan yang dibuat pada konfigurasi, Anda dapat membuat revisi konfigurasi. Untuk informasi selengkapnya, lihat [Creating and applying broker configurations](#).

## Atribut

Konfigurasi broker memiliki beberapa atribut, misalnya:

- Nama (`MyConfiguration`)
- ID (`c-1234a5b6-78cd-901e-2fgh-3i45j6k17819`)
- Amazon Resource Name (ARN) (`arn:aws:mq:us-east-2:123456789012:configuration:c-1234a5b6-78cd-901e-2fgh-3i45j6k17819`)

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](#)

## Pengguna

Pengguna ActiveMQ adalah orang atau aplikasi yang dapat mengakses antrian 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 mengirim ke, menerima dari, dan mengelola antrian 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 selengkapnya, lihat [Siklus hidup konfigurasi broker Amazon MQ](#).

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 dan mengelola pengguna broker ActiveMQ](#)
- [Pengguna](#)

### Atribut

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

- [ID Operasi REST: Pengguna](#)
- [ID Operasi REST: Pengguna](#)

### Penyimpanan

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	<ul style="list-style-type: none"> <li>• Tanda kutip saham</li> <li>• Pembaruan data lokasi</li> <li>• Data yang sering diubah</li> </ul>	5.000	Tidak ada
Amazon EBS	Tetap	<ul style="list-style-type: none"> <li>• Volume teks yang tinggi</li> <li>• Proses pemesanan</li> </ul>	500	Beberapa salinan dalam satu Availability Zone (AZ)
Amazon EFS	Tetap	Transaksi keuangan	80	Beberapa salinan di beberapa AZ

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

## Arsitektur broker

Amazon MQ untuk broker ActiveMQ dapat dibuat sebagai broker satu contoh atau pilang 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-INJAK](#)
- STOMP berakhir [WebSocket](#)

### Topik

- [Broker instans tunggal Amazon MQ](#)
- [Broker aktif/siaga Amazon MQ untuk ketersediaan tinggi](#)
- [Jaringan broker Amazon MQ](#)

## Broker instans tunggal Amazon MQ

SEBUAH broker satu contoh 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 (AZ). Amazon EBS menyediakan penyimpanan tingkat blok yang dioptimalkan untuk latensi rendah dan throughput tinggi. Untuk informasi selengkapnya tentang opsi penyimpanan, lihat [Storage](#).

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

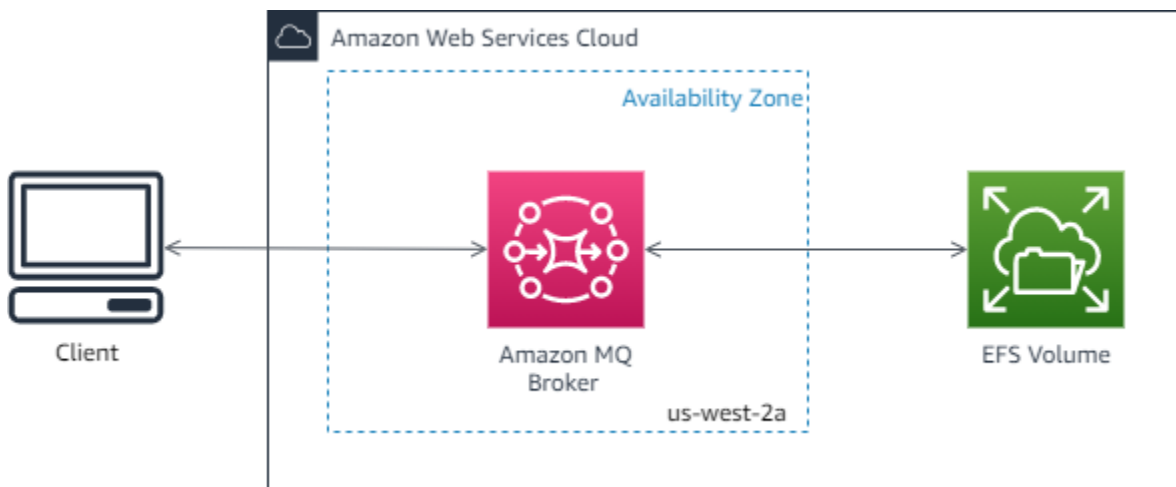
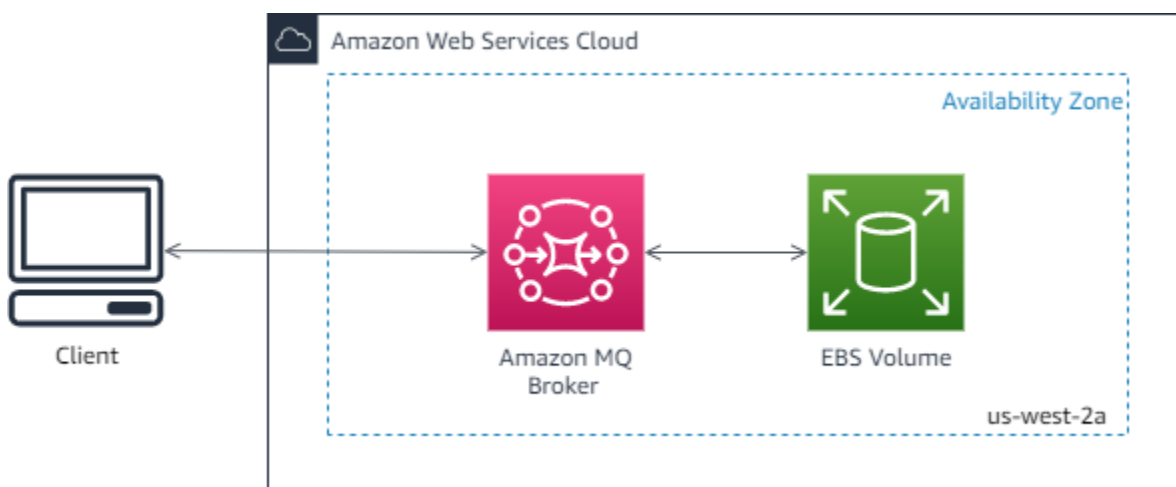


Diagram berikut menggambarkan broker instans tunggal dengan penyimpanan Amazon EBS yang direplikasi di beberapa server dalam satu AZ.



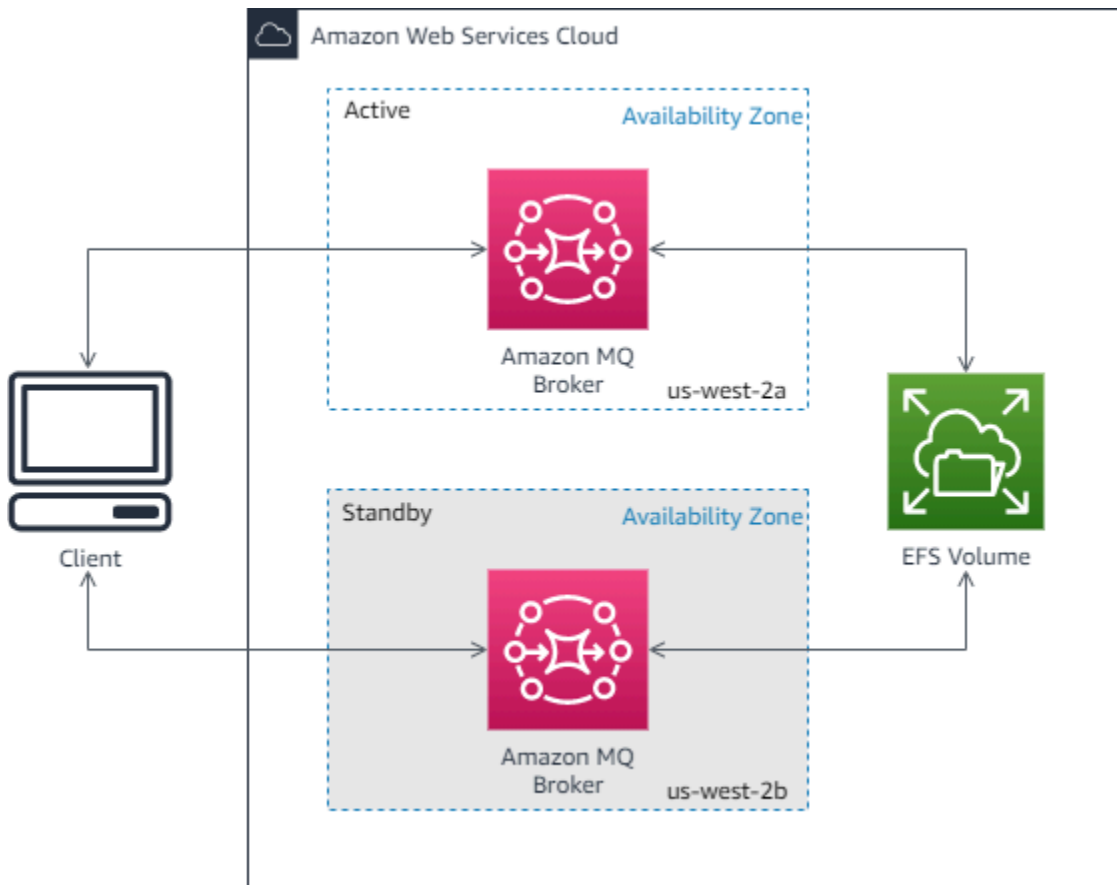
## Broker aktif/siaga Amazon MQ untuk ketersediaan tinggi

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 (AZ). 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. Ketika Anda mem-boot ulang broker, failover hanya berlangsung beberapa detik.

Untuk broker aktif/siaga, Amazon MQ menyediakan dua URL Konsol Web ActiveMQ, tetapi hanya satu URL 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 wire, Anda dapat mengizinkan aplikasi untuk terhubung ke titik akhir menggunakan [Transportasi Failover](#).

Diagram berikut menggambarkan broker aktif/siaga dengan penyimpanan Amazon EFS direplikasi di beberapa AZ.



## Jaringan broker Amazon MQ

Amazon MQ mendukung fitur jaringan broker ActiveMQ.

Jaringan broker terdiri dari beberapa [broker instans tunggal](#) aktif atau [broker aktif/siaga](#). Anda dapat mengkonfigurasi jaringan broker dalam berbagai [topologi](#) (misalnya, konsentrator,, pohon hub-and-spokes, atau mesh), tergantung pada kebutuhan aplikasi Anda, seperti ketersediaan tinggi dan skalabilitas. Sebagai contoh, jaringan [hub-and-spoke](#) broker dapat meningkatkan ketahanan, menjaga pesan jika salah satu broker tidak dapat dijangkau. Jaringan broker dengan topologi [konsentrator](#) dapat mengumpulkan pesan dari sejumlah besar broker, menerima pesan masuk, dan mengonsentrasikannya ke broker yang lebih terpusat, untuk menangani beban dari banyaknya pesan masuk dengan lebih baik.

Untuk tutorial dan detail informasi konfigurasi, lihat hal berikut:

- [Creating and Configuring a Network of Brokers](#)
- [Mengonfigurasi Jaringan Broker dengan Benar](#)
- [networkConnector](#)

- [networkConnectionStartasinkron](#)
- [Jaringan Pialang](#) dalam dokumentasi ActiveMQ

Berikut adalah manfaat penggunaan jaringan broker:

- Membuat jaringan broker memungkinkan Anda meningkatkan throughput agregat serta jumlah maksimum koneksi produsen dan konsumen dengan menambahkan instans broker.
- Anda dapat memastikan ketersediaan yang lebih baik dengan mengizinkan produsen dan konsumen untuk mengetahui beberapa instans broker aktif. Ini memungkinkan mereka terhubung kembali ke instans baru jika salah satu instans yang saat ini terhubung menjadi tidak tersedia.
- Karena produsen dan konsumen dapat segera terhubung kembali ke simpul lain dalam jaringan broker, dan karena tidak perlu menunggu instans broker siaga untuk dipromosikan, koneksi ulang klien dalam jaringan broker lebih cepat dibandingkan untuk [broker aktif/siaga untuk ketersediaan tinggi](#).

Topik

- [Bagaimana cara kerja jaringan broker?](#)
- [Bagaimana Cara Jaringan Broker Menangani Kredensial?](#)
- [Cetak biru sampel](#)
- [Jaringan topologi broker](#)
- [Lintas wilayah](#)
- [Failover Dinamis dengan Konektor Transportasi](#)

Bagaimana cara kerja jaringan broker?

Amazon MQ mendukung fitur jaringan broker ActiveMQ dalam berbagai cara. Pertama, Anda dapat mengedit parameter dalam setiap konfigurasi broker untuk membuat jaringan broker, seperti yang Anda lakukan dengan ActiveMQ native. Kedua, Amazon MQ memiliki contoh cetak biru yang menggunakan AWS CloudFormation untuk mengotomatisasi pembuatan jaringan broker. Anda dapat men-deploy cetak biru sampel ini secara langsung dari konsol Amazon MQ, atau mengedit template AWS CloudFormation terkait untuk membuat topologi dan konfigurasi Anda sendiri.

Jaringan broker dibuat dengan menghubungkan satu broker ke broker lain menggunakan konektor jaringan. Setelah terhubung, broker ini memberikan penerusan pesan. Misalnya, jika Broker1 membuat konektor jaringan ke Broker2, pesan di Broker1 diteruskan ke Broker2 jika ada konsumen

pada untuk antrean atau topik pada broker tersebut. Jika konektor jaringan dikonfigurasi sebagai duplex, pesan juga diteruskan dari Broker2 ke Broker1. Konektor jaringan dikonfigurasi dalam broker konfigurasi. Lihat, [Konfigurasi](#). Misalnya, berikut ini adalah contoh `networkConnector` entri dalam konfigurasi broker:

```
<networkConnectors>
  <networkConnector name="connector_1_to_2" userName="myCommonUser" duplex="true"
    uri="static:(ssl://b-1234a5b6-78cd-901e-2fgh-3i45j6k17819-1.mq.us-
east-2.amazonaws.com:61617)"/>
</networkConnectors>
```

Jaringan broker memastikan bahwa pesan mengalir dari satu instans broker ke instans broker lainnya, meneruskan pesan hanya ke instans broker dengan konsumen yang sesuai. Untuk kepentingan instans broker yang berdekatan satu sama lain dalam jaringan, ActiveMQ mengirim pesan ke topik penasihat berisi produsen dan konsumen yang terhubung ke dan memutuskan koneksi dari jaringan. Ketika instans broker menerima informasi tentang konsumen yang mengonsumsi dari tujuan tertentu, instans broker mulai meneruskan pesan. Untuk informasi selengkapnya, lihat [Topik Penasihat](#) dalam dokumentasi ActiveMQ.

### 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 pengguna 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](#).

Broker harus berada dalam VPC yang sama atau VPC yang di-peer. Untuk informasi selengkapnya, lihat [Prasyarat](#) di tutorial [Creating and Configuring a Network of Brokers](#).

## Cetak biru sampel


Untuk mulai menggunakan Jaringan Broker, Amazon MQ menyediakan cetak biru sampel. Cetak biru sampel ini membuat deployment Jaringan Broker, dan semua sumber daya terkait menggunakan AWS CloudFormation. Dua sampel cetak biru yang tersedia adalah:

1. Jaringan mesh broker instans tunggal
2. Jaringan mesh dari broker aktif/siaga

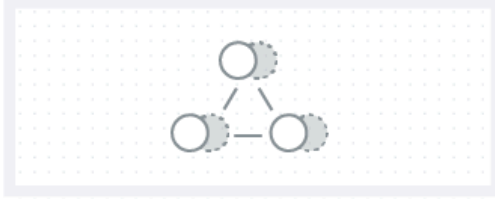
### Sample blueprints for a network of brokers

Networks of brokers provide high availability and scalability, and are suitable for production workloads. These sample blueprints use AWS CloudFormation to automatically deploy a network of brokers in the specific topology. [Info](#)

Mesh network of single-instance brokers  
Set of 3 single-instance brokers connected in a mesh network.



Mesh network of active/standby brokers  
Set of 3 active/standby brokers connected in a mesh network. Each broker has automatic failover capability to a standby in another AZ.



Dari halaman Buat broker, pilih salah satu cetak biru sampel dan pilih Selanjutnya. Setelah sumber daya dibuat, tinjau broker yang dihasilkan dan konfigurasi mereka di konsol Amazon MQ.

Dengan membuat broker dan mengonfigurasi elemen `networkConnector` yang berbeda dalam konfigurasi broker, Anda dapat membuat jaringan broker di berbagai topologi. Untuk informasi selengkapnya tentang mengonfigurasi jaringan broker, lihat [Jaringan Broker](#) dalam dokumentasi ActiveMQ.

### Jaringan topologi broker

Dengan men-deploy broker, lalu mengonfigurasi entri `networkConnector` dalam konfigurasinya, Anda dapat membangun jaringan broker menggunakan topologi jaringan yang berbeda. Konektor jaringan menyediakan penerusan pesan sesuai permintaan di antara broker yang terhubung. Koneksi dapat dikonfigurasi sebagai dupleks, di mana pesan diteruskan secara dua arah di antara broker, atau tidak dupleks, di mana penerusan hanya menyebar dari satu broker ke broker lainnya. Misalnya,



jika kita memiliki koneksi dupleks antara Broker1 dan Broker2, pesan akan diteruskan dari satu broker ke broker lain jika ada konsumen.



Dengan konektor jaringan dupleks, pesan diteruskan dari satu broker ke broker lainnya. Pesan diteruskan sesuai permintaan: jika ada konsumen di Broker2 untuk pesan di Broker1, pesan akan diteruskan. Demikian pula, jika ada konsumen di Broker1 untuk pesan di Broker2, pesan juga akan diteruskan.

Untuk koneksi nondupleks, pesan diteruskan hanya dari satu broker ke broker lainnya. Dalam contoh ini, jika ada konsumen di Broker2 untuk pesan di Broker1, pesan juga akan diteruskan. Tapi pesan tidak akan diteruskan dari Broker2 ke Broker1.



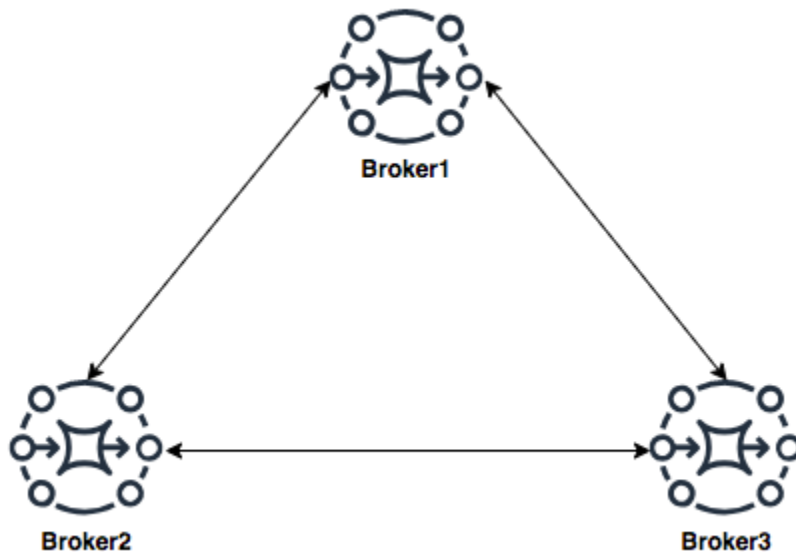
Menggunakan kedua konektor jaringan dupleks dan nondupleks, Anda dapat membangun jaringan broker di sejumlah topologi jaringan.

#### **Note**

Dalam setiap contoh topologi jaringan, elemen `networkConnector` mereferensikan titik akhir dari broker yang terhubung. Mengganti entri titik akhir broker di atribut `uri` dengan titik akhir broker Anda. Lihat, [Listing brokers and viewing broker details](#).

## Topologi mesh

Topologi mesh menyediakan beberapa broker yang semuanya saling terhubung. Contoh sederhana ini menghubungkan tiga broker instans tunggal, tetapi Anda dapat mengonfigurasi lebih banyak broker sebagai mesh.



Topologi ini, dan topologi yang menyertakan mesh pasangan broker aktif/siaga, dapat dibuat menggunakan cetak biru sampel di konsol Amazon MQ. Anda dapat membuat deployment cetak biru sampel untuk melihat jaringan broker yang bekerja, dan meninjau caranya dikonfigurasi.

Anda dapat mengonfigurasi tiga jaringan mesh broker seperti ini dengan menambahkan konektor jaringan ke Broker1 yang membuat koneksi dupleks ke Broker2 dan Broker3, dan koneksi dupleks tunggal antara Broker2 dan Broker3.

Konektor jaringan untuk Broker1:

```
<networkConnectors>
  <networkConnector name="connector_1_to_2" userName="myCommonUser" duplex="true"
    uri="static:(ssl://b-987615k4-32ji-109h-8gfe-7d65c4b132a1-2.mq.us-
east-2.amazonaws.com:61617)"/>
  <networkConnector name="connector_1_to_3" userName="myCommonUser" duplex="true"
    uri="static:(ssl://b-743c885d-2244-4c95-af67-a85017ff234e-3.mq.us-
east-2.amazonaws.com:61617)"/>
</networkConnectors>
```

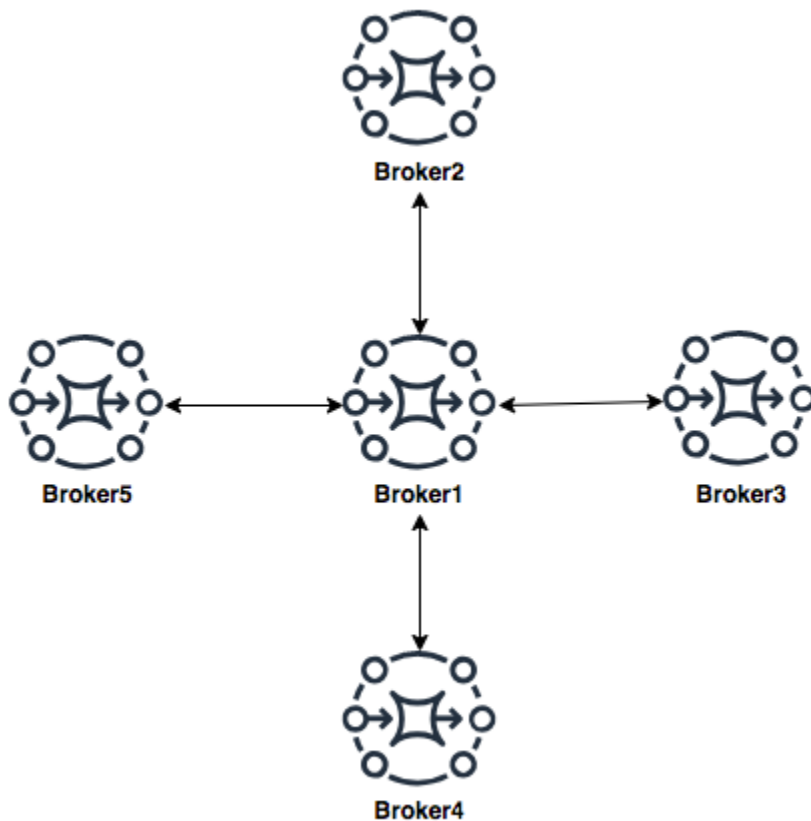
Konektor jaringan untuk Broker2:

```
<networkConnectors>
  <networkConnector name="connector_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>
```

Dengan menambahkan konektor di atas ke konfigurasi Broker1 dan Broker2, Anda dapat membuat mesh di antara ketiga broker tersebut yang meneruskan pesan di antara semua broker sesuai permintaan. Untuk informasi selengkapnya, lihat [Amazon MQ Broker Configuration Parameters](#).

### Topologi Hub and Spoke

Dalam topologi hub and spoke, pesan disimpan jika ada gangguan pada broker apapun di spoke. Pesan diteruskan seluruhnya, dan hanya Broker1 sentral yang sangat penting untuk operasi jaringan.

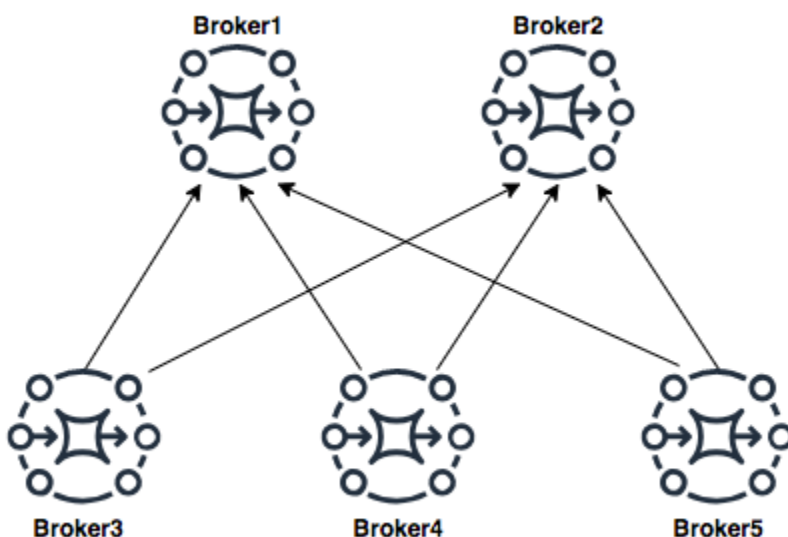


Untuk mengonfigurasi jaringan broker hub and spoke dalam contoh ini, Anda dapat menambahkan `networkConnector` ke tiap broker di spoke dalam konfigurasi Broker1.

```
<networkConnectors>
  <networkConnector name="connector_hub_and_spoke_2" userName="myCommonUser"
duplex="true"
  uri="static:(ssl://b-1234a5b6-78cd-901e-2fgh-3i45j6k17819-1.mq.us-
east-2.amazonaws.com:61617)"/>
  <networkConnector name="connector_hub_and_spoke_3" userName="myCommonUser"
duplex="true"
  uri="static:(ssl://b-987615k4-32ji-109h-8gfe-7d65c4b132a1-2.mq.us-
east-2.amazonaws.com:61617)"/>
  <networkConnector name="connector_hub_and_spoke_4" userName="myCommonUser"
duplex="true"
  uri="static:(ssl://b-743c885d-2244-4c95-af67-a85017ff234e-3.mq.us-
east-2.amazonaws.com:61617)"/>
  <networkConnector name="connector_hub_and_spoke_5" userName="myCommonUser"
duplex="true"
  uri="static:(ssl://b-62a7fb31-d51c-466a-a873-905cd660b553-4.mq.us-
east-2.amazonaws.com:61617)"/>
</networkConnectors>
```

## Topologi konsentrator

Dalam contoh topologi ini, tiga broker di bagian bawah dapat menangani sejumlah besar koneksi, dan pesan tersebut dikonsentrasikan ke Broker1 dan Broker2. Setiap broker lainnya memiliki koneksi nondupleks ke broker yang lebih sentral. Untuk menskalakan kapasitas topologi ini, Anda dapat menambahkan broker tambahan yang menerima pesan dan mengonsentrasikan pesan tersebut di Broker1 dan Broker2.



Untuk mengonfigurasi topologi ini, setiap broker di bagian bawah akan berisi konektor jaringan untuk setiap broker tempat pesan dikonsentrasikan.

Konektor jaringan untuk Broker3:

```
<networkConnectors>
  <networkConnector name="3_to_1" userName="myCommonUser" duplex="false"
    uri="static:(ssl://b-1234a5b6-78cd-901e-2fgh-3i45j6k17819-1.mq.us-
    east-2.amazonaws.com:61617)"/>
  <networkConnector name="3_to_2" userName="myCommonUser" duplex="false"
    uri="static:(ssl://b-987615k4-32ji-109h-8gfe-7d65c4b132a1-2.mq.us-
    east-2.amazonaws.com:61617)"/>
</networkConnectors>
```

Konektor jaringan untuk Broker4:

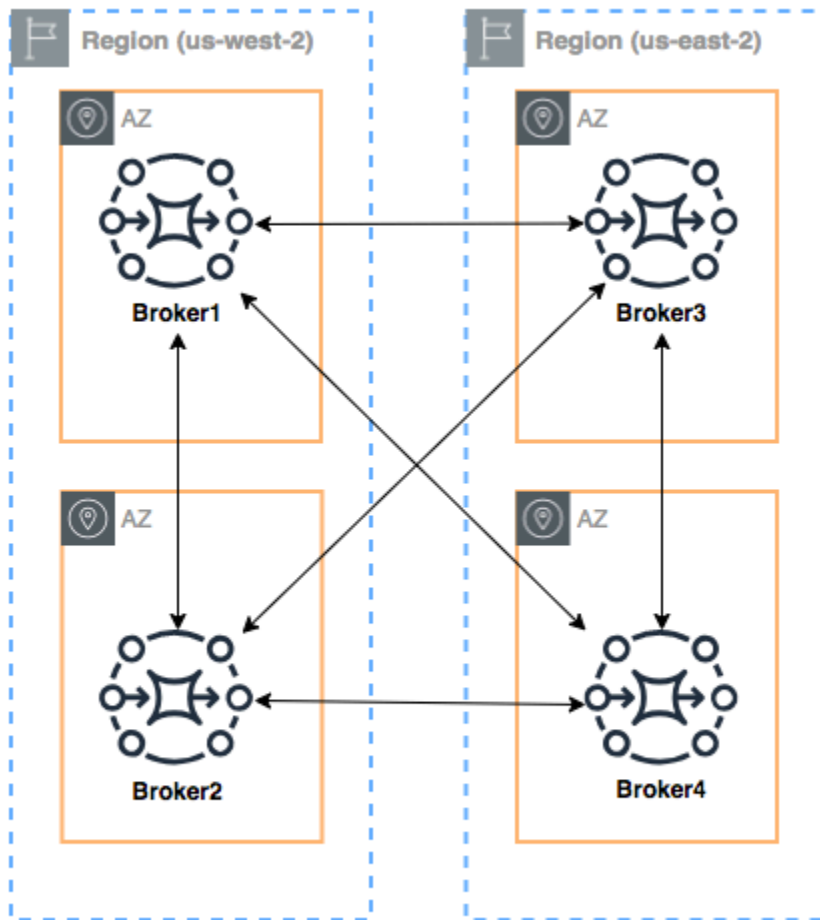
```
<networkConnectors>
  <networkConnector name="4_to_1" userName="myCommonUser" duplex="false"
    uri="static:(ssl://b-1234a5b6-78cd-901e-2fgh-3i45j6k17819-1.mq.us-
    east-2.amazonaws.com:61617)"/>
  <networkConnector name="4_to_2" userName="myCommonUser" duplex="false"
    uri="static:(ssl://b-1234a5b6-78cd-901e-2fgh-3i45j6k17819-1.mq.us-
    east-2.amazonaws.com:61617)"/>
</networkConnectors>
```

Konektor jaringan untuk Broker5:

```
<networkConnectors>
  <networkConnector name="5_to_1" userName="myCommonUser" duplex="false"
    uri="static:(ssl://b-1234a5b6-78cd-901e-2fgh-3i45j6k17819-1.mq.us-
    east-2.amazonaws.com:61617)"/>
  <networkConnector name="5_to_2" userName="myCommonUser" duplex="false"
    uri="static:(ssl://b-987615k4-32ji-109h-8gfe-7d65c4b132a1-2.mq.us-
    east-2.amazonaws.com:61617)"/>
</networkConnectors>
```

## Lintas wilayah

Untuk mengonfigurasi jaringan broker yang mencakup wilayah AWS, deploy broker di wilayah tersebut dan konfigurasi 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-987615k4-32ji-109h-8gfe-7d65c4b132a1-2.mq.us-
east-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>
```

```
</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-987615k4-32ji-109h-8gfe-7d65c4b132a1-2.mq.us-
east-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-3i45j6k17819-1.mq.us-east-2.amazonaws.com:61617)
```

Ketika broker baru terhubung, broker akan menerima daftar URI dari semua broker dalam 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 [Ops Sisi Broker untuk Failover](#) dalam dokumentasi ActiveMQ.

## Amazon MQ untuk konfigurasi broker ActiveMQ

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

### Topik

- [Bekerja dengan file konfigurasi Spring XML](#)
- [Membuat, mengedit, dan menerapkan konfigurasi broker ActiveMQ](#)
- [Elemen yang Diizinkan dalam Konfigurasi Amazon MQ](#)
- [Elemen dan Atribut yang Diizinkan dalam Konfigurasi Amazon MQ](#)
- [Elemen, Elemen Pengumpulan Anak, dan Elemen Anaknya yang Diizinkan dalam Konfigurasi Amazon MQ](#)

## Bekerja dengan 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-5.17.3.xsd.zip](#)
- [amazon-mq-active-mq-5.17.2.xsd.zip](#)
- [amazon-mq-active-mq-5.17.1.xsd.zip](#)
- [amazon-mq-active-mq-5.16.5.xsd.zip](#)
- [amazon-mq-active-mq-5.16.4.xsd.zip](#)
- [amazon-mq-active-mq-5.16.3.xsd.zip](#)
- [amazon-mq-active-mq-5.16.2.xsd.zip](#)
- [amazon-mq-active-mq-5.15.15.xsd.zip](#)
- [amazon-mq-active-mq-5.15.14.xsd.zip](#)
- [amazon-mq-active-mq-5.15.13.xsd.zip](#)
- [amazon-mq-active-mq-5.15.12.xsd.zip](#)
- [amazon-mq-active-mq-5.15.10.xsd.zip](#)
- [amazon-mq-active-mq-5.15.9.xsd.zip](#)
- [amazon-mq-active-mq-5.15.8.xsd.zip](#)
- [amazon-mq-active-mq-5.15.6.xsd.zip](#)
- [amazon-mq-active-mq-5.15.0.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, mengedit, dan menerapkan 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.

Untuk informasi selengkapnya, lihat yang berikut:

- [Konfigurasi](#)
- [Siklus hidup konfigurasi broker Amazon MQ](#)
- [Amazon MQ Broker Configuration Parameters](#)

Contoh berikut menunjukkan cara membuat dan menerapkan konfigurasi broker Amazon MQ menggunakan AWS Management Console.

Topik

- [Buat Konfigurasi Baru](#)
- [Buat Revisi Konfigurasi Baru](#)
- [Terapkan Revisi Konfigurasi ke Broker Anda](#)
- [Mengedit Revisi Konfigurasi](#)

Buat Konfigurasi Baru

1. Masuk ke [konsol Amazon MQ](#).
2. Di sebelah kiri, perluas panel navigasi dan pilih Konfigurasi.

**Amazon MQ** ×

Brokers

**Configurations**

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

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

**Note**

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

Pada **MyConfiguration** halaman, jenis mesin broker dan versi yang digunakan revisi konfigurasi baru Anda (misalnya, Apache ActiveMQ 5.15.8) 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.

### Revision 1 Auto-generated default for MyBroker-configuration on ActiveMQ 5.15.0 Latest

Amazon MQ configurations support a limited subset of ActiveMQ properties. [Info](#)

```

1 <?xml version="1.0" encoding="UTF-8" standalone="yes"?>
2 <broker xmlns="http://activemq.apache.org/schema/core">
3   <!--
4     A configuration contains all of the settings for your ActiveMQ broker, in XML format
     (similar to ActiveMQ's activemq.xml file).
5     You can create a configuration before creating any brokers. You can then apply the
     configuration to one or more brokers.

```

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](#).

Pembuatan perubahan pada konfigurasi tidak akan segera menerapkan perubahan ke broker. Untuk menerapkan perubahan Anda, Anda harus menunggu jendela pemeliharaan berikutnya atau [reboot broker](#). Untuk informasi selengkapnya, lihat [Siklus hidup konfigurasi broker Amazon MQ](#).

Saat ini, Anda tidak dapat menghapus konfigurasi.

Terapkan Revisi Konfigurasi ke Broker Anda

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

**Amazon MQ** ×

Brokers

Configurations

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

**⚠ Important**

Broker Anda akan menjadi offline saat sedang di-boot ulang.

**5. Pilih Terapkan .**

Revisi konfigurasi Anda diterapkan ke broker pada waktu yang ditentukan.

**Mengedit Revisi Konfigurasi**

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

**ℹ Note**

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

Pada **MyBroker** halaman, jenis mesin broker dan versi 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.

**ℹ Note**

Mengedit konfigurasi saat ini membuat revisi konfigurasi baru.

**Revision 1** Auto-generated default for MyBroker-configuration on ActiveMQ 5.15.0 Latest

Amazon MQ configurations support a limited subset of ActiveMQ properties. [Info](#)

```
1 <?xml version="1.0" encoding="UTF-8" standalone="yes"?>
2 <broker xmlns="http://activemq.apache.org/schema/core">
3   <!--
4     A configuration contains all of the settings for your ActiveMQ broker, in XML format
     (similar to ActiveMQ's activemq.xml file).
5     You can create a configuration before creating any brokers. You can then apply the
     configuration to one or more brokers.
```

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](#).

Pembuatan perubahan pada konfigurasi tidak akan segera menerapkan perubahan ke broker. Untuk menerapkan perubahan Anda, Anda harus menunggu jendela pemeliharaan berikutnya atau [reboot broker](#). Untuk informasi selengkapnya, lihat [Siklus hidup konfigurasi broker Amazon MQ](#).

Saat ini, Anda tidak dapat menghapus konfigurasi.

## Elemen yang 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](#))

`fixedSizedSubscriptionRecoveryPolicy` ([atribut](#))

`forcePersistencyModeBrokerPlugin` ([atribut](#))

`individualDeadLetterStrategy` ([atribut](#))

## Elemen

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 [\(elemen pengumpulan anak\)](#)

roundRobinDispatchPolicy

sharedDeadLetterStrategy [\(atribut | elemen pengumpulan anak\)](#)

simpleDispatchPolicy



## Elemen

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

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
abortSlowConsumerStrategy	abortConnection
	checkPeriod
	ignoreNetworkConsumers
	maxSlowCount
	maxSlowDuration
authorizationEntry	admin
	queue
	read
	tempQueue

Elemen	Atribut
	tempTopic
	topic
	write
broker	advisorySupport
	allowTempAutoCreationOnSend
	cacheTempDestinations
	consumerSystemUsagePortion
	dedicatedTaskRunner
	deleteAllMessagesOnStartup
	keepDurableSubsActive
	enableMessageExpirationOnActiveDurableSubs
	maxPurgedDestinationsPerSweep
	maxSchedulerRepeatAllowed
	monitorConnectionSplits
	<a href="#">networkConnectorStartAsync</a>
	offlineDurableSubscriberTaskSchedule
	offlineDurableSubscriberTimeout
	persistenceThreadPriority
	persistent

Elemen	Atribut
	populateJMSXUserID
	producerSystemUsagePortion
	rejectDurableConsumers
	rollbackOnlyOnAsyncException
	schedulePeriodForDestinationPurge
	schedulerSupport
	splitSystemUsageForProducersConsumers
	taskRunnerPriority
	timeBeforePurgeTempDestinations
	useAuthenticatedPrincipalForJMSXUserID
	useMirroredQueues
	useTempMirroredQueues
	useVirtualDestSubs
	useVirtualDestSubsOnCreation
	useVirtualTopics
cachedMessageGroupMapFactory	cacheSize
compositeQueue	concurrentSend
	copyMessage

Elemen	Atribut
	forwardOnly
	name
	sendWhenNotMatched
compositeTopic	concurrentSend
	copyMessage
	forwardOnly
	name
conditionalNetworkBridgeFilterFactory	sendWhenNotMatched
	rateDuration
	rateLimit
	replayDelay
	replayWhenNoConsumers
constantPendingMessageLimit Strategy	selectorAware
	limit
discarding	deadLetterQueue
	enableAudit
	expiration



Didukung di

Apache ActiveMQ 5.16.x


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

Elemen	Atribut
	maxProducersToAudit
	processExpired
	processNonPersistent
	queuePrefix
	queueSuffix
	topicPrefix
	topicSuffix
	useQueueForQueueMessages
	useQueueForTopicMessages
messageGroupHashBucketFactory	bucketCount
mirroredQueue	cacheSize
	copyMessage
	postfix
oldestMessageEvictionStrategy	prefix
	evictExpiredMessagesHighWatermark
oldestMessageWithLowestPriorityEvictionStrategy	evictExpiredMessagesHighWatermark
policyEntry	advisoryForConsumed
	advisoryForDelivery
	advisoryForDiscardingMessages

Elemen	Atribut
	advisoryForFastProducers
	advisoryForSlowConsumers
	advisoryWhenFull
	allConsumersExclusiveByDefault
	alwaysRetroactive
	blockedProducerWarningInterval
	consumersBeforeDispatchStarts
	cursorMemoryHighWaterMark
	doOptimizeMessageStorage
	durableTopicPrefetch
	enableAudit
	expireMessagesPeriod
	gcInactiveDestinations
	gcWithNetworkConsumers
	inactiveTimeoutBeforeGC
	inactiveTimeoutBeforeGC
	includeBodyForAdvisory
	lazyDispatch
	maxAuditDepth
	maxBrowsePageSize



Elemen	Atribut
	maxDestinations
	maxExpirePageSize
	maxPageSize
	maxProducersToAudit
	maxQueueAuditDepth
	memoryLimit
	messageGroupMapFactoryType
	minimumMessageSize
	optimizedDispatch
	optimizeMessageStoreInFlightLimit
	persistJMSRedelivered
	prioritizedMessages
	producerFlowControl
	queue
	queueBrowserPrefetch
	queuePrefetch
	reduceMemoryFootprint
	sendAdvisoryIfNoConsumers
	sendFailIfNoSpace

Elemen	Atribut
	sendFailIfNoSpaceAfterTimeout <div style="border: 1px solid #add8e6; border-radius: 10px; padding: 10px; margin: 10px 0;">  Didukung di            Apache ActiveMQ 5.16.4 dan di atasnya         </div>
	sendDuplicateFromStoreToDLQ
	storeUsageHighWaterMark
	strictOrderDispatch
	tempQueue
	tempTopic
	timeBeforeDispatchStarts
	topic
	topicPrefetch
	useCache
	useConsumerPriority
usePrefetchExtension	
prefetchRatePendingMessageLimitStrategy	multiplier
queryBasedSubscriptionRecoveryPolicy	query
queue	DLQ
	physicalName

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

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

Elemen	Atribut
	physicalName
transportConnector	<ul style="list-style-type: none"> <li>• name</li> <li>updateClusterClients</li> <li>rebalanceClusterClients</li> <li>updateClusterClientsOnRemove</li> </ul>
uniquePropertyMessageEvictionStrategy	<ul style="list-style-type: none"> <li>evictExpiredMessagesHighWatermark</li> <li>propertyName</li> </ul>
virtualTopic	<ul style="list-style-type: none"> <li>concurrentSend</li> <li>local</li> <li>dropOnResourceLimit</li> <li>name</li> <li>postfix</li> <li>prefix</li> <li>selectorAware</li> <li>setOriginalDestination</li> <li>transactedSend</li> </ul>

## 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
<code>authorizationMap</code>	<code>authorizationEntries</code>	<a href="#">authorizationEntry</a>
		<code>tempDestinationAuthorizationEntry</code>

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	<a href="#">networkConnector</a>
	persistenceAdapter	<a href="#">kahaDB</a>
	plugins	authorizationPlugin
		discardingDLQBrokerPlugin
		forcePersistencyModeBrokerPlugin
		redeliveryPlugin

Elemen	Elemen Pengumpulan Anak	Elemen Anak
		statisticsBrokerPlugin
		timeStampingBrokerPlugin
	systemUsage	<a href="#">systemUsage</a>
	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 konfigurasi 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>
```

`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`

`nama`

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 adalah `duplex`, dan URI OpenWire titik akhir memiliki prefiks `static:`, menunjukkan one-to-one koneksi antara broker.

```
<networkConnectors>
  <networkConnector name="connector_1_to_2" userName="myCommonUser" duplex="true"
    uri="static:(ssl://b-1234a5b6-78cd-901e-2fgh-3i45j6k17819-1.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 adalah `duplex`, dan daftar URI OpenWire titik akhir yang dipisahkan koma memiliki prefiks failover antara broker. `masterslave`: 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-3i45j6k17819-1.mq.us-
east-2.amazonaws.com:61617,
    ssl://b-9876l5k4-32ji-109h-8gfe-7d65c4b132a1-2.mq.us-east-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 XML-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 sampah dan pembersihan tidak terjadi ketika broker tidak berfungsi, sehingga 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>
```

## Mengelola versi mesin Amazon MQ for ActiveMQ

Apache ActiveMQ mengatur nomor versi sesuai dengan spesifikasi versioning semantik sebagai `X.Y.Z`. Dalam implementasi Amazon MQ for ActiveMQ, `X.Y` menunjukkan versi utama, dan `Z` mewakili nomor versi minor. Amazon MQ menganggap perubahan versi sebagai utama jika nomor versi utama berubah. Contohnya, meningkatkan dari versi 5.15 ke 5.16 dianggap sebagai peningkatan versi utama. Perubahan versi dianggap minor jika hanya nomor versi minor yang berubah. Contohnya, meningkatkan dari versi 5.15.14 ke 5.15.15 dianggap sebagai peningkatan versi minor.

Amazon MQ for ActiveMQ saat ini mendukung versi mesin Apache ActiveMQ berikut.

Versi utama	Versi minor
ActiveMQ 5.17	<ul style="list-style-type: none"> <li>• 5.17.6 (disarankan)</li> </ul>
ActiveMQ 5.16	<ul style="list-style-type: none"> <li>• 5.16.7</li> </ul>
ActiveMQ 5.15	<ul style="list-style-type: none"> <li>• 5.15.16</li> </ul>

Versi minor berikut telah usang.

Versi minor yang tidak digunakan lagi

Versi utama	Versi minor
ActiveMQ 5.17	<ul style="list-style-type: none"> <li>• 5.17.3</li> <li>• 5.17.2</li> <li>• 5.17.1</li> </ul>
ActiveMQ 5.16	<ul style="list-style-type: none"> <li>• 5.16.5</li> <li>• 5.16.4</li> <li>• 5.16.3</li> <li>• 5.16.2</li> </ul>
ActiveMQ 5.15	<ul style="list-style-type: none"> <li>• 5.15.15</li> <li>• 5.15.14</li> <li>• 5.15.13</li> <li>• 5.15.12</li> <li>• 5.15.10</li> <li>• 5.15.9</li> <li>• 5.15.8</li> <li>• 5.15.6</li> <li>• 5.15.0</li> </ul>

Ketika membuat broker baru Amazon MQ for ActiveMQ, Anda dapat menentukan versi mesin ActiveMQ yang didukung. Jika Anda menggunakan AWS Management Console untuk membuat broker, nomor versi mesin terbaru dijadikan default secara otomatis oleh Amazon MQ. Jika menggunakan AWS CLI atau API Amazon MQ untuk membuat broker, Anda memerlukan nomor versi mesin. Jika Anda tidak memberikan nomor versi, operasi akan menghasilkan pengecualian. Untuk mempelajari selengkapnya, lihat [create-broker](#) di Refensi Perintah AWS CLI dan [CreateBroker](#) dalam Referensi REST API Amazon MQ.

## Topik

- [Pemutakhiran versi utama dan minor](#)
- [Membuat daftar versi mesin yang didukung](#)

## Pemutakhiran versi utama dan minor

Dengan Amazon MQ, Anda dapat mengontrol waktu untuk meningkatkan broker ke versi baru. Ketika [upgrade versi minor otomatis](#) diaktifkan, Amazon MQ akan secara otomatis meningkatkan mesin broker Anda ke versi minor baru saat dirilis dan didukung oleh Amazon MQ.

Untuk melakukan upgrade versi utama, Anda harus secara manual meng-upgrade nomor versi mesin broker Anda. [Peningkatan versi minor dan mayor terjadi bersamaan dengan operasi patching broker lainnya, selama jendela pemeliharaan terjadwal Anda.](#) Jika Anda memilih keluar dari upgrade versi minor otomatis, Anda dapat secara manual meningkatkan broker Anda ke versi minor baru yang didukung dengan mengikuti prosedur yang sama sebagai upgrade besar.

Untuk informasi lebih lanjut tentang memperbarui preferensi broker Anda untuk mengaktifkan atau menonaktifkan peningkatan versi minor, dan meningkatkan broker Anda secara manual, lihat [the section called “Meningkatkan versi mesin”](#)

## Membuat daftar versi mesin yang didukung

Anda dapat membuat daftar semua versi mesin minor dan utama yang didukung menggunakan perintah AWS CLI [describe-broker-instance-options](#).

```
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 bagi ActiveMQ, dan tipe instans `mq.m5.large`, ganti *engine-type* dengan `ACTIVEMQ` dan *instance-type* dengan `mq.m5.large`.

## Contoh kerja menggunakan Java Message Service (JMS) dengan ActiveMQ

Contoh berikut menunjukkan cara bekerja dengan ActiveMQ secara pemrograman:


- TheOpenWirecontoh 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

1. Masuk ke [konsol Amazon MQ](#).
2. Dari daftar broker, pilih nama broker Anda (misalnya, MyBroker).
3. Pada **MyBroker** halaman, di Koneksibagian, perhatikan 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>5.15.8</version>
  </dependency>
  <dependency>
    <groupId>org.apache.activemq</groupId>
    <artifactId>activemq-pool</artifactId>
    <version>5.15.8</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.

## AmazonMQExample.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 JMSEException
{
    // 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);
    }
}
```



```
}
```

## 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 broker ActiveMQ](#)
- [Membuat dan mengonfigurasi jaringan broker Amazon MQ](#)
- [Menghubungkan aplikasi Java ke broker Amazon MQ Anda](#)
- [Mengintegrasikan broker ActiveMQ dengan LDAP](#)
- [Membuat dan mengelola pengguna broker ActiveMQ](#)

## Membuat dan mengonfigurasi broker ActiveMQ

Broker adalah lingkungan broker pesan yang berjalan di Amazon MQ. Ini adalah blok bangunan dasar Amazon MQ. Deskripsi gabungan dari instans broker kelas (m5, t3) dan ukuran (large, micro) adalah tipe instans broker (misalnya, mq.m5.large). Untuk informasi selengkapnya, lihat [Pialang](#).

Tugas Amazon MQ yang pertama dan paling umum adalah membuat broker. Contoh berikut menunjukkan bagaimana Anda dapat membuat dan mengkonfigurasi broker menggunakan AWS Management Console.


### Topik

- [Langkah 1: Mengonfigurasi Pengaturan Broker Dasar](#)
- [Langkah 2: \(Opsional\) Mengonfigurasi Pengaturan Broker Tambahan](#)
- [Langkah 3: Selesai Membuat Broker](#)
- [Mengedit versi mesin broker, jenis instans, CloudWatchlog, dan preferensi pemeliharaan](#)

## Langkah 1: Mengonfigurasi Pengaturan 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 [Broker Architecture](#).
    - SEBUAHPialang satu contohterdiri 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 [Broker 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 [Broker aktif/siaga Amazon MQ untuk ketersediaan tinggi](#).
    - Untuk informasi selengkapnya tentang cetak biru sampel jaringan broker, lihat [Cetak biru sampel](#).
  - 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 orang lainAWSlayanan, termasukCloudWatchLog. 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](#).
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 orang lain AWS layanan, termasuk CloudWatch Log. Nama pengguna broker tidak dimaksudkan untuk digunakan untuk data pribadi atau sensitif.

## Langkah 2: (Opsional) Mengonfigurasi Pengaturan Broker Tambahan

 Important

- Subnet – Broker instans tunggal memerlukan satu subnet (misalnya, subnet default). Broker aktif/siaga memerlukan dua subnet.
- Grup keamanan – Broker instans tunggal dan broker aktif/siaga memerlukan setidaknya satu grup keamanan (misalnya, grup keamanan default).
- VPC – Subnet dan grup keamanan broker harus berada dalam VPC yang sama. Sumber daya EC2-Classic tidak didukung. Amazon MQ hanya mendukung penghunian VPC default, dan tidak mendukung penghunian VPC khusus.
- Enkripsi – Pilih kunci utama pelanggan untuk mengenkripsi data Anda. Lihat [Enkripsi saat istirahat](#).
- Aksesibilitas publik – Menonaktifkan aksesibilitas publik membuat broker hanya dapat diakses dalam VPC Anda. Untuk informasi lebih lanjut, lihat [Lebih memilih broker tanpa aksesibilitas publik](#) dan [Mengakses konsol web broker tanpa aksesibilitas publik](#).

1. Perluas bagian Pengaturan tambahan.
2. Di bagian Konfigurasi, pilih Buat konfigurasi baru dengan nilai default atau Pilih konfigurasi yang ada. Untuk informasi lebih lanjut, lihat [Konfigurasi](#) dan [Amazon MQ Broker Configuration Parameters](#).
3. Di Log bagian, pilih apakah akan mempublikasikan Umumlog dan Auditlog ke Amazon CloudWatch Log. Untuk informasi selengkapnya, lihat [Configuring Amazon MQ to publish logs to Amazon CloudWatch Logs](#).


**⚠ 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.

Jika Anda tidak [mengonfigurasi kebijakan berbasis sumber daya untuk Amazon MQ](#), broker tidak dapat mempublikasikan log ke CloudWatch Log.

4. Di Bagian jaringan dan keamanan, konfigurasi konektivitas broker Anda:
  - a. Lakukan salah satu dari berikut ini:
    - Pilih Gunakan VPC, subnet, dan grup keamanan default.
    - Pilih Pilih VPC, subnet, dan grup keamanan yang ada.
      1. Jika memilih opsi ini, Anda dapat membuat Virtual Private Cloud (VPC) pada konsol Amazon VPC, memilih VPC yang ada, atau memilih VPC default. Untuk informasi selengkapnya, lihat [Apa yang dimaksud dengan Amazon VPC?](#) dalam Panduan Pengguna Amazon VPC.
      2. Setelah membuat atau memilih VPC, Anda dapat membuat Subnet baru di konsol Amazon VPC atau memilih subnet yang ada. Untuk informasi lebih lanjut, lihat [VPC dan Subnet](#) di Panduan Pengguna Amazon VPC.
      3. Setelah membuat atau memilih subnet, Anda dapat memilih Grup keamanan.
  - b. Pilih kunci utama pelanggan (CMK) yang akan digunakan untuk mengenkripsi data Anda. Lihat [Enkripsi saat istirahat](#).
  - c. Pilih Aksesibilitas publik broker Anda.
5. Di bagian Pemeliharaan, konfigurasi jadwal pemeliharaan broker Anda:

- a. Untuk meningkatkan broker ke versi baru saat dirilis Apache, 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).

 Note

Untuk broker aktif/siaga, jika salah satu instans broker dalam pemeliharaan, Amazon MQ memerlukan beberapa saat untuk mengeluarkan instans tidak aktif dari layanan. Hal ini memungkinkan instans siaga sehat untuk menjadi aktif dan mulai menerima komunikasi masuk.

- b. Lakukan salah satu dari berikut ini:
  - Untuk mengizinkan Amazon MQ memilih jendela pemeliharaan secara otomatis, pilih Tidak ada preferensi.
  - Untuk mengatur jendela pemeliharaan kustom, pilih Pilih jendela pemeliharaan lalu tentukan Mulai hari dan Waktu mulaipeningkatan.


### Langkah 3: Selesai Membuat Broker

1. Pilih 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.

	Name ▼	Status ▼	Deployment mode ▼	Instance type ▼
	MyBroker	Running	Single-instance broker	mq.m5.large

2. Pilih **MyBroker**.

Pada **MyBroker** halaman, diHubungkan bagian, perhatikan broker Anda [Konsol web ActiveMQ URL](#), misalnya:

```
https://b-1234a5b6-78cd-901e-2fgh-3i45j6k17819-1.mq.us-east-2.amazonaws.com:8162
```

Perhatikan juga [Titik Akhir protokol tingkat wire](#). Berikut ini adalah contoh dari OpenWire titik akhir:

```
ssl://b-1234a5b6-78cd-901e-2fgh-3i45j6k17819-1.mq.us-east-2.amazonaws.com:61617
```

### Note

Untuk broker aktif/siaga, Amazon MQ menyediakan dua URL Konsol Web ActiveMQ, tetapi hanya satu URL 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 [Broker Architecture](#).

Untuk titik akhir protokol tingkat wire, Anda dapat mengizinkan aplikasi untuk terhubung ke titik akhir menggunakan [Transportasi Failover](#).

## Mengedit versi mesin broker, jenis instans, CloudWatchlog, dan preferensi pemeliharaan

Selain [mengedit konfigurasi broker dan mengelola revisi konfigurasi](#), Anda dapat mengonfigurasi preferensi khusus untuk broker.

### Note

Semua preferensi kecuali untuk peningkatan versi minor otomatis mengharuskan Anda untuk menjadwalkan modifikasi. Untuk informasi selengkapnya, lihat [Siklus hidup konfigurasi broker Amazon MQ](#).


Contoh berikut menunjukkan cara mengedit preferensi broker Amazon MQ ActiveMQ menggunakan AWS Management Console.

### Mengedit opsi broker ActiveMQ

1. Masuk ke [konsol Amazon MQ](#).

2. Dari daftar broker, pilih broker Anda (misalnya, MyBroker) dan kemudian pilih Sunting.
3. Pada Sunting **MyBroker** halaman, di Spesifikasi bagian, pilih Versi mesin broker atau Jenis Instance Broker.
4. Di bagian Konfigurasi, pilih konfigurasi dan revisi untuk broker Anda. Untuk informasi selengkapnya, lihat [Creating and applying broker configurations](#).
5. Di bagian Jaringan dan keamanan, pilih grup dari drop-down Grup keamanan, atau pilih Buat grup keamanan baru untuk membuka konsol Amazon VPC.
6. Di CloudWatch Log bagian, pilih apakah akan mempublikasikan Umum log dan Audit log ke Amazon CloudWatch Log.

Untuk informasi lebih lanjut tentang mengkonfigurasi CloudWatch log untuk broker ActiveMQ, lihat [Configuring Amazon MQ to publish logs to Amazon CloudWatch Logs](#).


 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.

Jika Anda tidak [mengonfigurasi kebijakan berbasis sumber daya untuk Amazon MQ](#), broker tidak dapat mempublikasikan log ke CloudWatch Log.

7. Di bagian Pemeliharaan, konfigurasi jadwal pemeliharaan broker Anda:

Untuk meningkatkan broker ke versi baru saat dirilis AWS, 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).

 Note

Untuk broker aktif/siaga, jika salah satu instans broker dalam pemeliharaan, Amazon MQ memerlukan beberapa saat untuk mengeluarkan instans tidak aktif dari layanan. Hal ini memungkinkan instans siaga sehat untuk menjadi aktif dan mulai menerima komunikasi masuk.

8. Pilih Jadwalkan perubahan.

**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.

## Membuat dan mengonfigurasi jaringan broker Amazon MQ

Jaringan broker terdiri dari beberapa [broker instans tunggal](#) aktif atau [broker aktif/siaga](#). Anda dapat mengonfigurasi jaringan broker dalam berbagai [topologi](#) (Sebagai contoh, konsentrator, hub-and-spokes, bagan, atau jala), tergantung pada kebutuhan aplikasi Anda, seperti ketersediaan tinggi dan skalabilitas. Sebagai contoh, jaringan [hub-and-spoke](#) broker dapat meningkatkan ketahanan, menjaga pesan jika salah satu broker tidak dapat dijangkau. Jaringan broker dengan topologi [konsentrator](#) dapat mengumpulkan pesan dari sejumlah besar broker, menerima pesan masuk, dan mengonsentrasikannya ke broker yang lebih terpusat, untuk menangani beban dari banyaknya pesan masuk dengan lebih baik. 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 [Creating and configuring a broker](#).
- Kedua broker harus berada dalam VPC yang sama atau VPC yang di-peer. Untuk informasi lebih lanjut tentang VPC, lihat [Apa yang dimaksud dengan Amazon VPC?](#) dalam Panduan Pengguna Amazon VPC dan [Apa itu Peering VPC?](#) dalam Panduan Peering Amazon VPC.

### 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 dan mengelola 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. Pada [Konsol Amazon MQ](#), pada `MyBroker2` halaman, di `Detail` bagian, 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. Di `Edit` aturan masuk kotak dialog, tambahkan aturan untuk `OpenWire` titik akhir.
  - a. Pilih `Tambahkan aturan`.
  - b. Untuk `Jenis`, pilih `TCP Kustom`.
  - c. Untuk `Rentang Pelabuhan`, ketik `OpenWire` pelabuhan (`61617`).
  - d. Lakukan salah satu dari 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, untuk `Sumber`, 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 `MyBroker1` halaman, pilih `Sunting`.
- b. Pada `SuntingMyBroker1` halaman, di `Konfigurasi` bagian, 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 dari berikut:
  - Jika Anda menghubungkan broker ke broker satu contoh, gunakan `static:awalan` dan `OpenWire` titik akhir `uri` untuk `MyBroker2`. Misalnya:

```
<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` + `failoverTransportasi` dan `OpenWire` titik akhir `uri` untuk kedua broker dengan parameter kueri berikut `?randomize=false&maxReconnectAttempts=0`. Misalnya:

```
<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-
east-2.amazonaws.com:61617)?randomize=false&maxReconnectAttempts=0)"/>
</networkConnectors>
```

 Note

Jangan sertakan kredensial 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 MyBroker1 halaman, pilih Sunting.
  - b. Pada Sunting MyBroker1 halaman, di Konfigurasi bagian, 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 bahwa Anda [aktifkan koneksi masuk](#) dari mesin lokal Anda untuk broker MyBroker1 pada port 8162 (untuk ActiveMQ Web Console) dan port 61617 (untuk OpenWire titik akhir).

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. Misalnya:

```
activemq consumer --brokerUrl "ssl://
b-1234a5b6-78cd-901e-2fgh-3i45j6k17819-1.mq.us-east-2.amazonaws.com:61617" \
--user commonUser \
--password myPassword456 \
--destination queue://MyQueue
```

Konsumen terhubung keOpenWiretitik akhir dariMyBroker2dan mulai mengonsumsi pesan dari antrianMyQueue.

5. Dari setiap mesin yang memiliki akses ke broker MyBroker1, buat produsen dan kirimkan beberapa pesan. Misalnya:

```
activemq producer --brokerUrl "ssl://
b-987615k4-32ji-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 keOpenWiretitik akhir dariMyBroker1dan mulai menghasilkan pesan persisten untuk mengantriMyQueue.

## 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 antrian, 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

1. Masuk ke [konsol Amazon MQ](#).
2. Dari daftar broker, pilih nama broker Anda (misalnya, MyBroker).
3. Pada **MyBroker** halaman, di Koneksi bagian, perhatikan 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>5.15.8</version>
  </dependency>
  <dependency>
    <groupId>org.apache.activemq</groupId>
    <artifactId>activemq-pool</artifactId>
    <version>5.15.8</version>
  </dependency>
</dependencies>
```

Untuk informasi selengkapnya tentang `activemq-client.jar`, lihat [Konfigurasi Awal](#) dalam 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

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

**ℹ Note**

Untuk broker aktif/siaga, Amazon MQ menyediakan dua URL Konsol Web ActiveMQ, tetapi hanya satu URL 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 [Broker Architecture](#).

Untuk titik akhir protokol tingkat wire, Anda dapat mengizinkan aplikasi untuk terhubung ke titik akhir menggunakan [Transportasi Failover](#).

```
// 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, antrian 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 produser.**

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

## Untuk Membuat Konsumen 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, antrian 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 olahpesan AWS (seperti Amazon SQS), konsumen terus-menerus terhubung ke broker.

4. Menutup konsumen, sesi, dan koneksi.

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

## Mengintegrasikan broker ActiveMQ dengan LDAP

 Important

Integrasi LDAP tidak didukung untuk broker RabbitMQ.


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 konfigurasi 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 antrian 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 antrian untuk grup atau pengguna tertentu melalui server LDAP. Anda melakukannya dengan membuat entri yang mewakili topik dan antrian di server LDAP lalu menetapkan izin untuk pengguna atau grup LDAP tertentu. Kemudian Anda dapat mengonfigurasi broker untuk mengambil data otorisasi dari server LDAP.

## 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.

## Authentication and Authorization

Simple Authentication and Authorization  
Authenticate and authorize users using the credentials stored in a broker.

LDAP Authentication and Authorization  
Authenticate and authorize users using the credentials stored in an LDAP server.

Provide details for your organization's Active Directory or other LDAP server. [Info](#)

Fully qualified domain name

example.com

*optional second server name*

Service account username

Fully qualified name of the user that opens the connection to the directory server.

myserviceaccount

Service account password

The password for the service account provided above.

Maximum of 128 characters

Show

### LDAP login configuration

Your server configuration to search and authenticate users.

User Base

Fully qualified name of the directory where you want to search for users.

ou=user, dc=example, dc=com

User Search Matching

The search criteria for the user object applied to the directory provided above.

(uid=0)

Role Base

Fully qualified name of the directory to search for a user's groups.

ou=user, dc=example, dc=com

Role Search Matching

The search criteria for the group object applied to the directory provided above.

(uid=0)

► Optional settings

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.



## Authentication and Authorization

Simple Authentication and Authorization  
Authenticate and authorize users using the credentials stored in a broker.

LDAP Authentication and Authorization  
Authenticate and authorize users using the credentials stored in an LDAP server.

Provide details for your organization's Active Directory or other LDAP server. [Info](#)

Fully qualified domain name

example.com

*optional second server name*

Service account username

Fully qualified name of the user that opens the connection to the directory server.

myserviceaccount

Service account password

The password for the service account provided above.

Maximum of 128 characters

Show

### LDAP login configuration

Your server configuration to search and authenticate users.

User Base

Fully qualified name of the directory where you want to search for users.

ou=user, dc=example, dc=com

User Search Matching

The search criteria for the user object applied to the directory provided above.

(uid=0)

Role Base

Fully qualified name of the directory to search for a user's groups.

ou=user, dc=example, dc=com

Role Search Matching

The search criteria for the group object applied to the directory provided above.

(uid=0)

► Optional settings

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.

**Role Search Matching**

The search criteria for the group object applied to the directory provided above.

```
(member:=uid={1})
```

**▼ Optional settings****User Role Name**

Specifies the name of the LDAP attribute for the user group membership.

**Role Name**

Specifies the LDAP attribute that identifies the group name attribute in the object returned from the group membership query.

 **User Search Subtree**

This defines the directory search scope for the user. If set to true, scope is to search the entire sub-tree.

 **Role Search Subtree**

This defines the directory search scope for the role/group. If set to true, scope is to search the entire sub-tree.

## Cara kerja integrasi LDAP

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

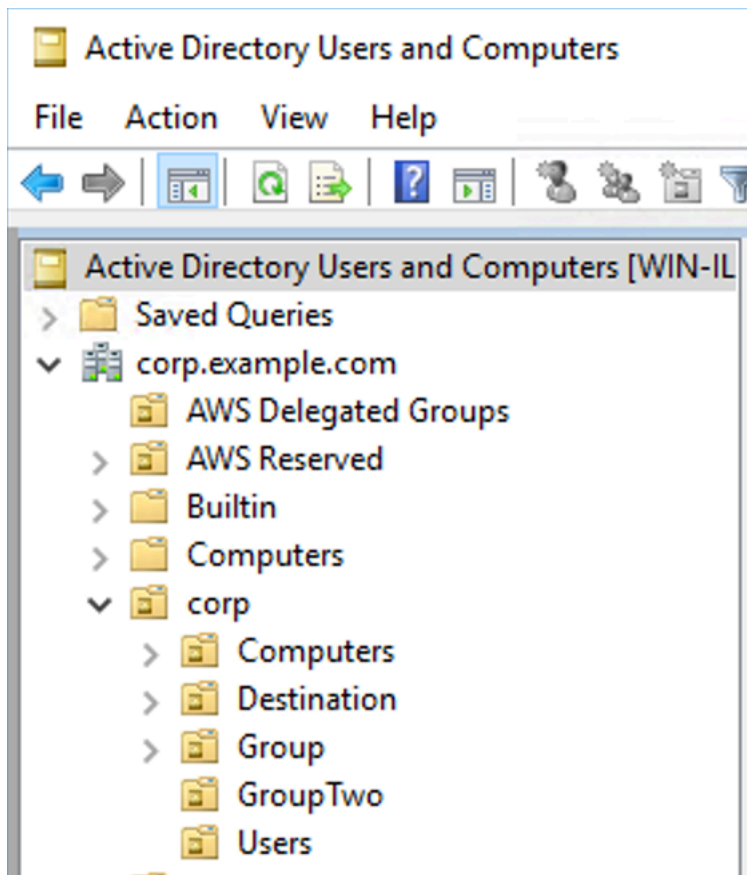
### Autentikasi

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 `corp`, `example`, dan `com`, lalu di dalamnya Anda memiliki unit organisasi `corp` dan `Users`, Anda akan menggunakan hal berikut sebagai basis pengguna:

```
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 elemen `cachedLDAPAuthorizationMap` di file konfigurasi `activemq.xml` broker, Anda harus memilih opsi Autentikasi dan Otorisasi LDAP saat [membuat konfigurasi melalui AWS Management Console](#), atau mengatur properti [authenticationStrategy](#) ke LDAP saat membuat konfigurasi baru menggunakan API Amazon MQ.

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`

Ketika Anda membuat broker, Amazon MQ menggantikan nilai yang Anda berikan melalui AWS Management Console, atau di properti [ldapServerMetadata](#) permintaan API Anda, untuk atribut di atas.

Hal berikut mendemonstrasikan contoh kerja `cachedLdapAuthorizationMap`.

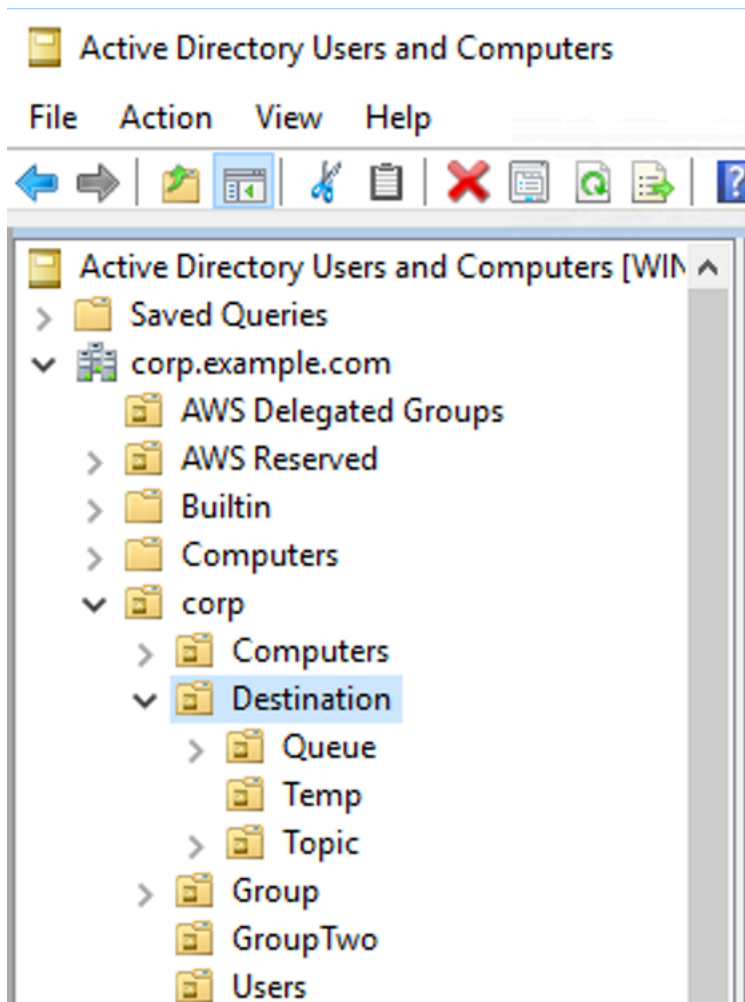
```
<authorizationPlugin>
  <map>
    <cachedLDAPAuthorizationMap
```

```
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, gunakan komponen domain `corp`, `example`, dan `com` yang sama, Anda akan menentukan unit organisasi bernama `destination` untuk menampung 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
```

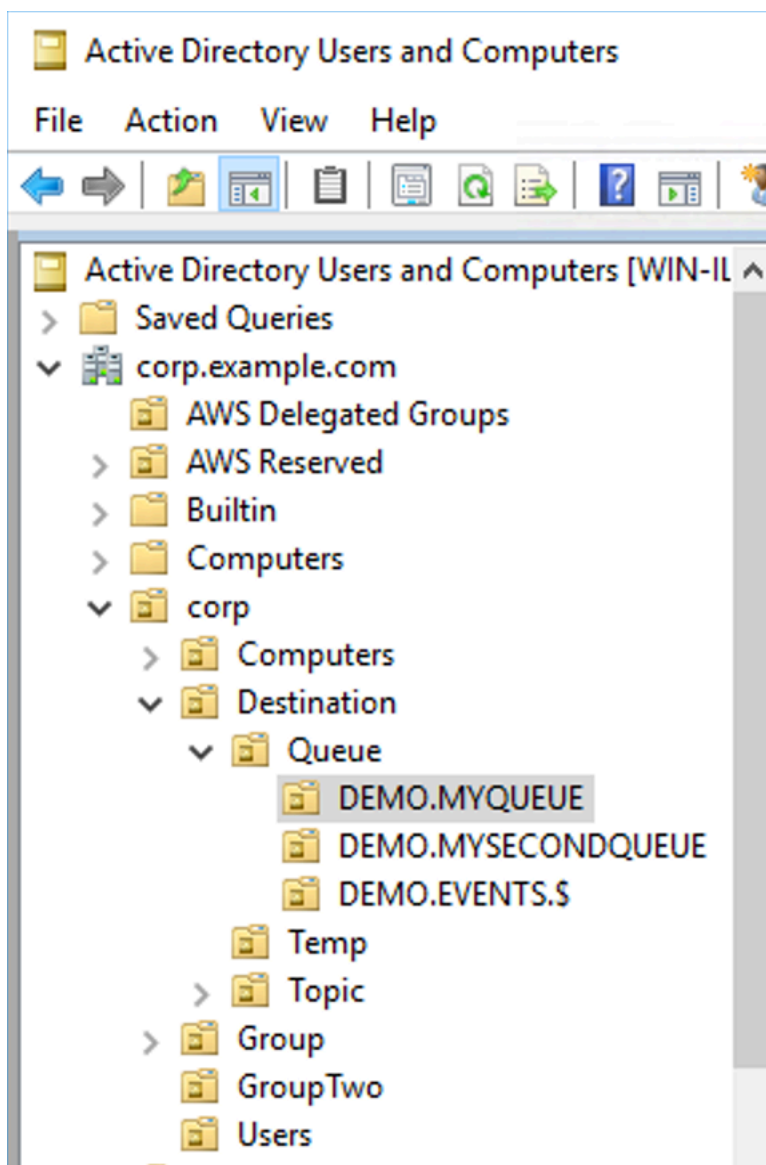
**Note**

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
```

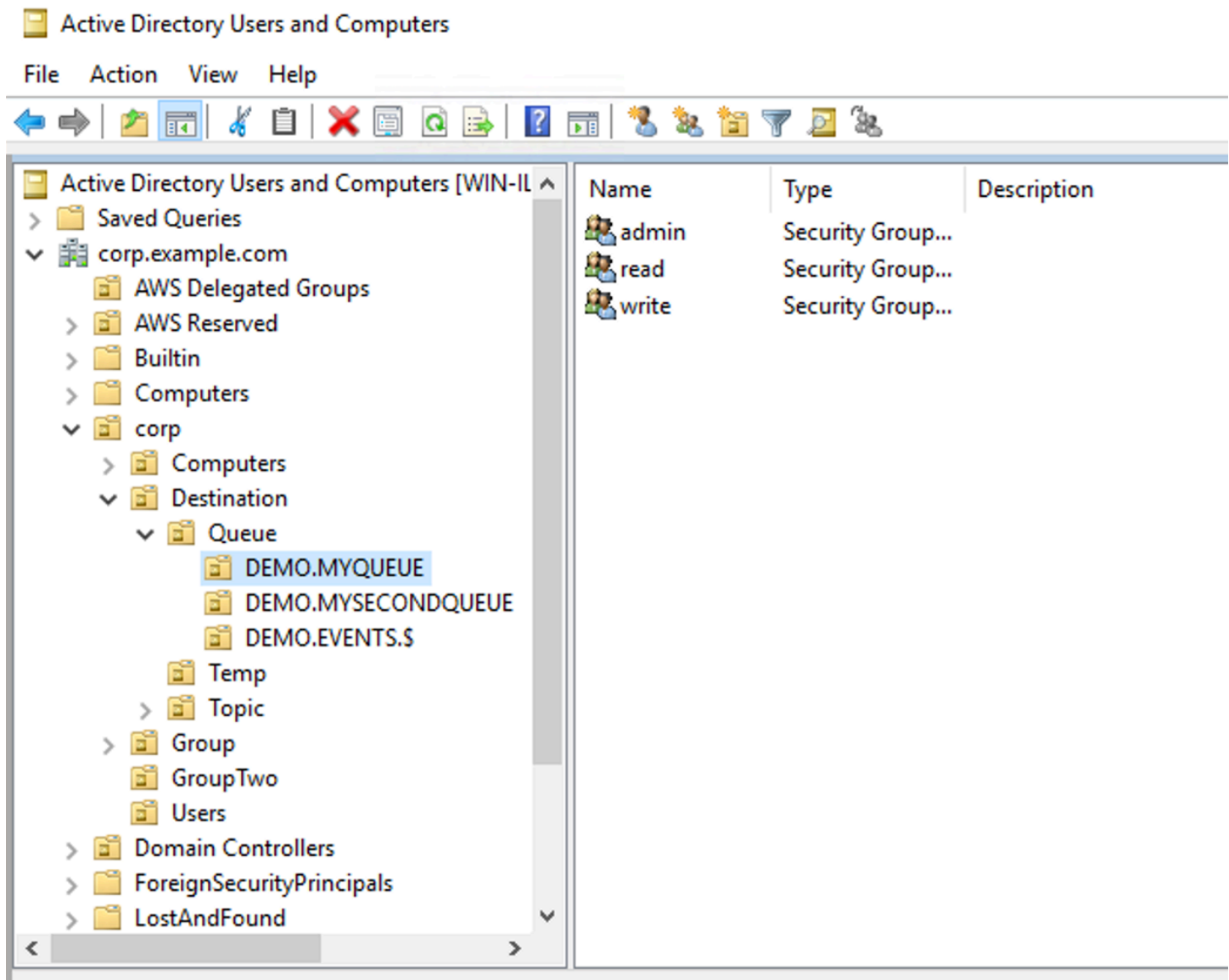




## Grup Keamanan

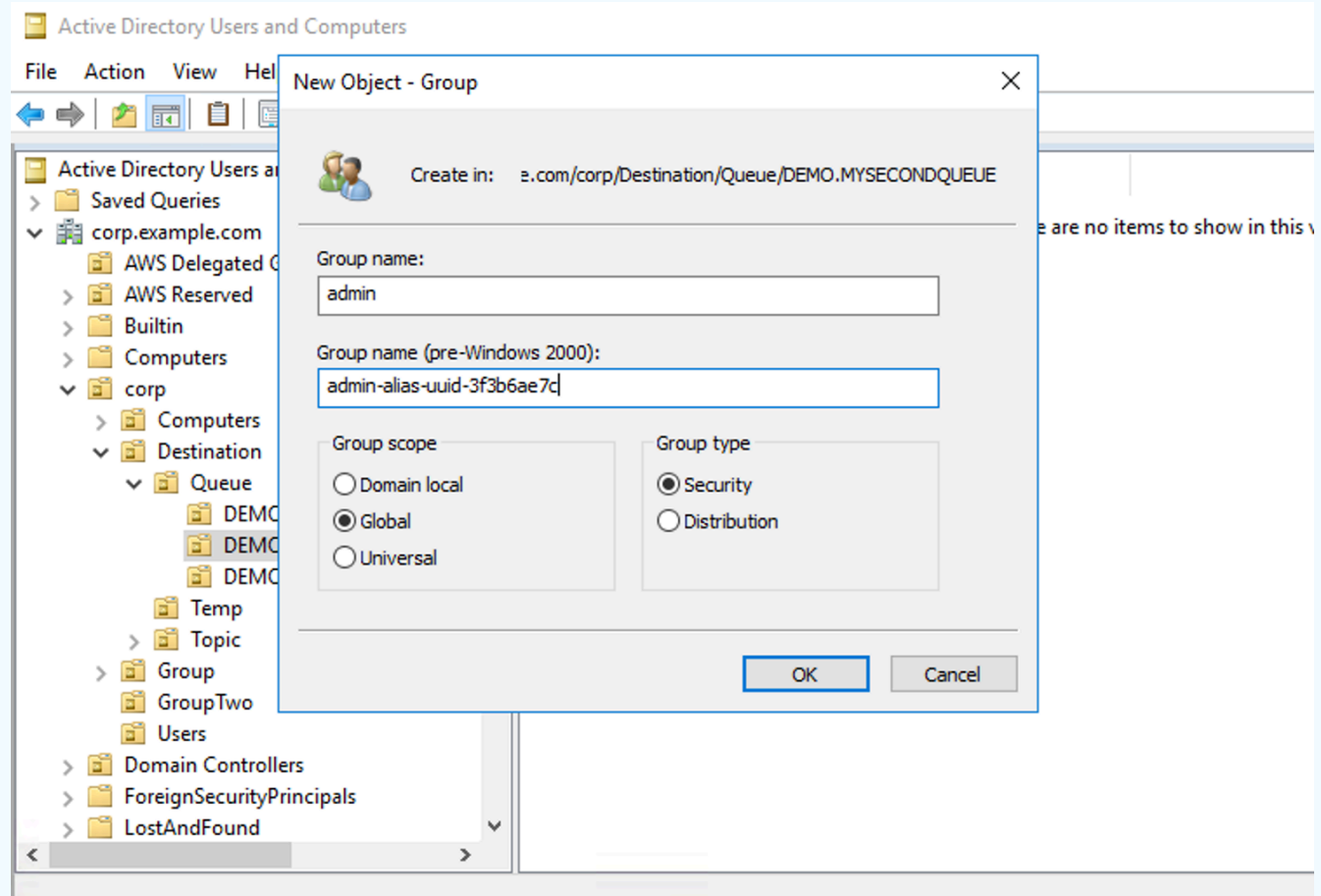
Dalam setiap OU yang mewakili tujuan atau wildcard, Anda harus membuat tiga grup keamanan. Seperti semua izin di ActiveMQ, ini adalah izin baca/tulis/admin. Untuk informasi selengkapnya tentang hal yang dapat dilakukan pengguna dengan setiap izin tersebut, lihat [Keamanan](#) dalam dokumentasi ActiveMQ.

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 \(dan berikutnya\)](#).

## Membuat dan mengelola pengguna broker ActiveMQ

Pengguna ActiveMQ adalah orang atau aplikasi yang dapat mengakses antrian 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 mengirim ke, menerima dari, dan mengelola antrian 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.

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

Topik

- [Untuk membuat pengguna baru](#)
- [Untuk mengedit pengguna yang sudah ada](#)
- [Untuk menghapus pengguna yang sudah ada](#)

Untuk membuat pengguna 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 `Pengguna` bagian, semua pengguna untuk broker ini terdaftar.

	Username	Console access	Groups	Pending modifications
<input type="radio"/>	paolo.santos	No	Devs	
<input type="radio"/>	jane.doe	Yes	Admins	

- Pilih Buat pengguna.
- Di kotak dialog Buat pengguna, ketik Nama pengguna dan Kata sandi.
- (Opsional) Ketik nama grup tempat pengguna berada, dipisahkan dengan koma (misalnya: Devs, Admins).
- (Opsional) Untuk mengizinkan pengguna mengakses [Konsol Web ActiveMQ](#), pilih Konsol Web ActiveMQ.
- 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](#). Untuk informasi selengkapnya, lihat [Siklus hidup konfigurasi broker Amazon MQ](#).

## Untuk mengedit pengguna yang sudah ada

- Masuk ke [konsol Amazon MQ](#).
- Dari daftar broker, pilih nama broker Anda (misalnya, MyBroker) dan kemudian pilih Lihat detail.

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

	Username	Console access	Groups	Pending modifications
<input type="radio"/>	paolo.santos	No	Devs	
<input type="radio"/>	jane.doe	Yes	Admins	

- Pilih kredensi masuk Anda dan pilih Sunting.
- Kotak dialog Edit pengguna akan ditampilkan.

- (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](#). Untuk informasi selengkapnya, lihat [Siklus hidup konfigurasi broker Amazon MQ](#).

Untuk menghapus pengguna yang sudah ada

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 Pengguna bagian, semua pengguna untuk broker ini terdaftar.

	Username	Console access	Groups	Pending modifications
<input type="radio"/>	paolo.santos	No	Devs	
<input type="radio"/>	jane.doe	Yes	Admins	

3. Pilih kredensi login Anda (misalnya, **MyUser**) dan kemudian 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](#). Untuk informasi selengkapnya, lihat [Siklus hidup konfigurasi broker Amazon MQ](#).

## Praktik terbaik Amazon MQ for 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.

Topik

- [Terhubung ke Amazon MQ](#)
- [Memastikan performa Amazon MQ yang efektif](#)
- [Hindari mulai ulang lambat dengan memulihkan transaksi XA yang disiapkan](#)

## Terhubung ke Amazon MQ

Pola desain berikut dapat meningkatkan efektivitas koneksi aplikasi Anda ke broker Amazon MQ Anda.

Topik

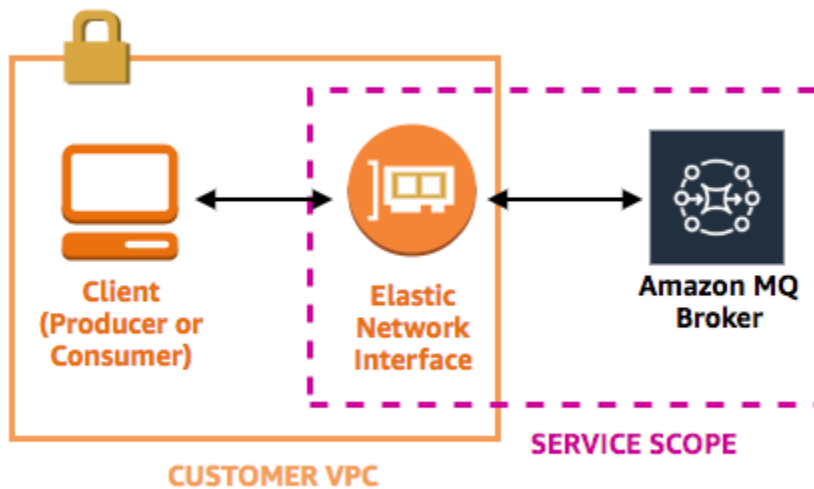
- [Jangan Pernah Memodifikasi atau Menghapus Antarmuka Jaringan Elastis Amazon MQ](#)
- [Selalu Gunakan Pooling Koneksi](#)
- [Selalu Gunakan Transportasi Failover untuk Terhubung ke Beberapa Titik Akhir Broker](#)
- [Hindari Penggunaan Penyeleksi Pesan](#)
- [Memilih Tujuan Virtual untuk Langganan Tahan Lama](#)
- [Jika menggunakan peering Amazon VPC, hindari IP klien dalam kisaran CIDR10.0.0.0/16](#)

## 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 [Getting Started with Amazon MQ](#)), 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. Sebagai contoh:

```
failover:(ssl://b-1234a5b6-78cd-901e-2fgh-3i45j6k17819-1.mq.us-
east-2.amazonaws.com:61617,ssl://b-9876l5k4-32ji-109h-8gfe-7d65c4b132a1-2.mq.us-
east-2.amazonaws.com:61617)?randomize=true
```



## 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 Amazon VPC, hindari IP klien dalam kisaran **CIDR10.0.0.0/16**

Jika Anda menyiapkan peering Amazon VPC antara infrastruktur di lokasi dan broker Amazon MQ Anda, Anda tidak boleh mengonfigurasi koneksi klien dengan IP dalam kisaran CIDR10.0.0.0/16.

## Memastikan performa Amazon MQ yang efektif

Pola desain berikut dapat meningkatkan efektivitas dan performa broker Amazon MQ Anda.

Topik

- [Menonaktifkan Penyimpanan dan Pengiriman Bersamaan untuk Antrean dengan Konsumen Lambat](#)
- [Memilih Tipe Instans Broker yang Tepat untuk Throughput Terbaik](#)
- [Pilih jenis penyimpanan broker yang tepat untuk throughput terbaik](#)
- [Mengonfigurasi Jaringan Broker dengan Benar](#)

## 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](#), konfigurasi 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 disiapkan dan belum terselesaikan, Anda dapat menggunakan `JournalFilesForFastRecovery` metrik pada 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.setUser_name(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()`.

## Replikasi data Lintas Wilayah untuk Amazon MQ untuk ActiveMQ

Amazon MQ for 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.

Bagian ini menyediakan tutorial tentang cara mengatur replikasi data Lintas Wilayah dengan Amazon MQ untuk ActiveMQ.

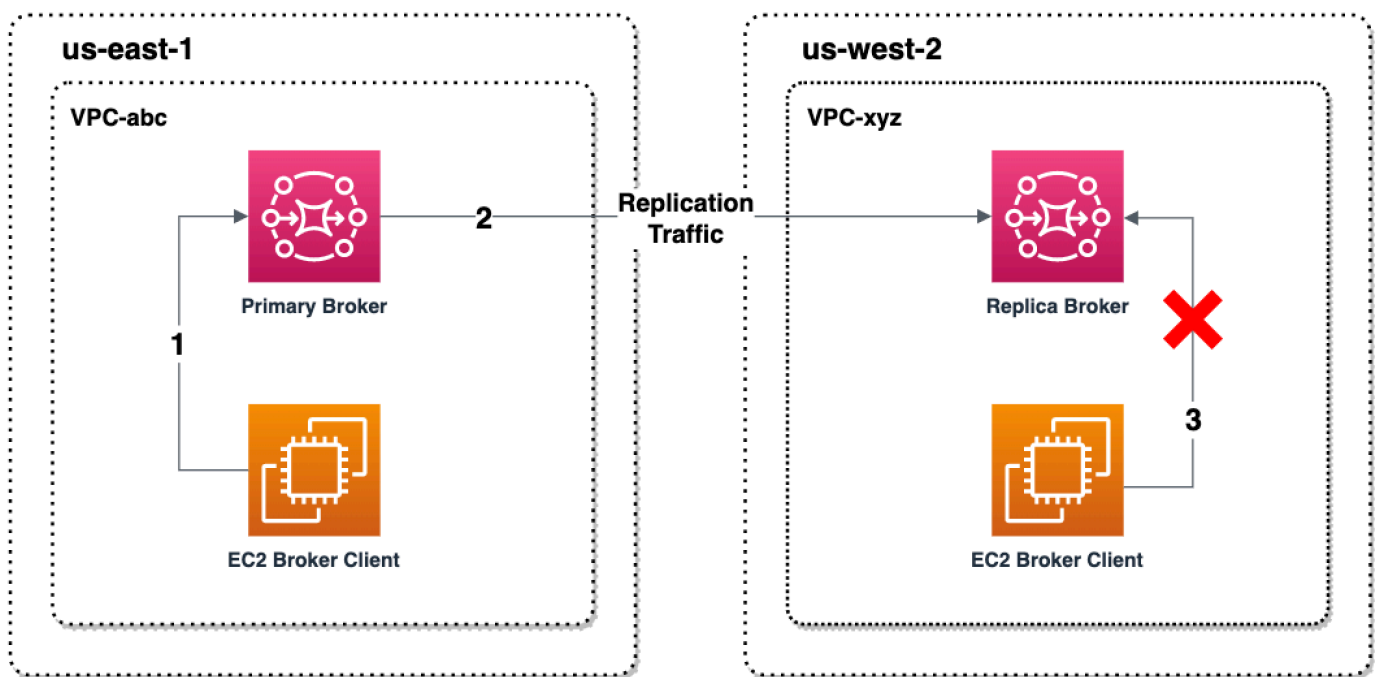
## Topik

- [Pialang primer dan replika di Amazon MQ](#)
- [Membuat dan menghapus broker replikasi data Lintas Wilayah](#)
- [Memulai switchover atau failover untuk mempromosikan broker replika ke peran broker utama](#)
- [Metrik replikasi data lintas wilayah di Amazon CloudWatch](#)

## Pialang primer dan replika di Amazon MQ

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 pasangan broker aktif/siaga berlebihan yang disebut sebagai broker utama. Wilayah sekunder terdiri dari sepasang broker aktif/siaga berlebihan yang disebut sebagai broker replika.

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



Broker primer dan replika bertindak sebagai solusi pemulihan data lintas wilayah. Jika broker utama di Wilayah utama 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 primer dan replika, lihat [Membuat dan menghapus broker replikasi data Lintas Wilayah](#).

#### Note

Hanya tersedia untuk broker aktif/siaga.

## Membuat dan menghapus broker replikasi data Lintas Wilayah

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 API Amazon `Promote MQ`. Untuk informasi lebih lanjut tentang broker primer dan replika, lihat [Pialang primer dan replika di Amazon MQ](#).

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](#)
- [Hapus broker CRDR](#)

### 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 [Broker Architecture](#).
5. Pilih Next (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.

Konfigurasi 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 [Membuat dan menghubungkan ke broker ActiveMQ](#)

## Hapus broker CRDR

Untuk menghapus broker CRDR primer atau replika, Anda harus terlebih dahulu memutuskan pasangan kemudian 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 “unpair” 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, lihat [Deleting a broker](#).

## Memulai switchover atau failover untuk mempromosikan broker replika 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 failover 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 failover 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.

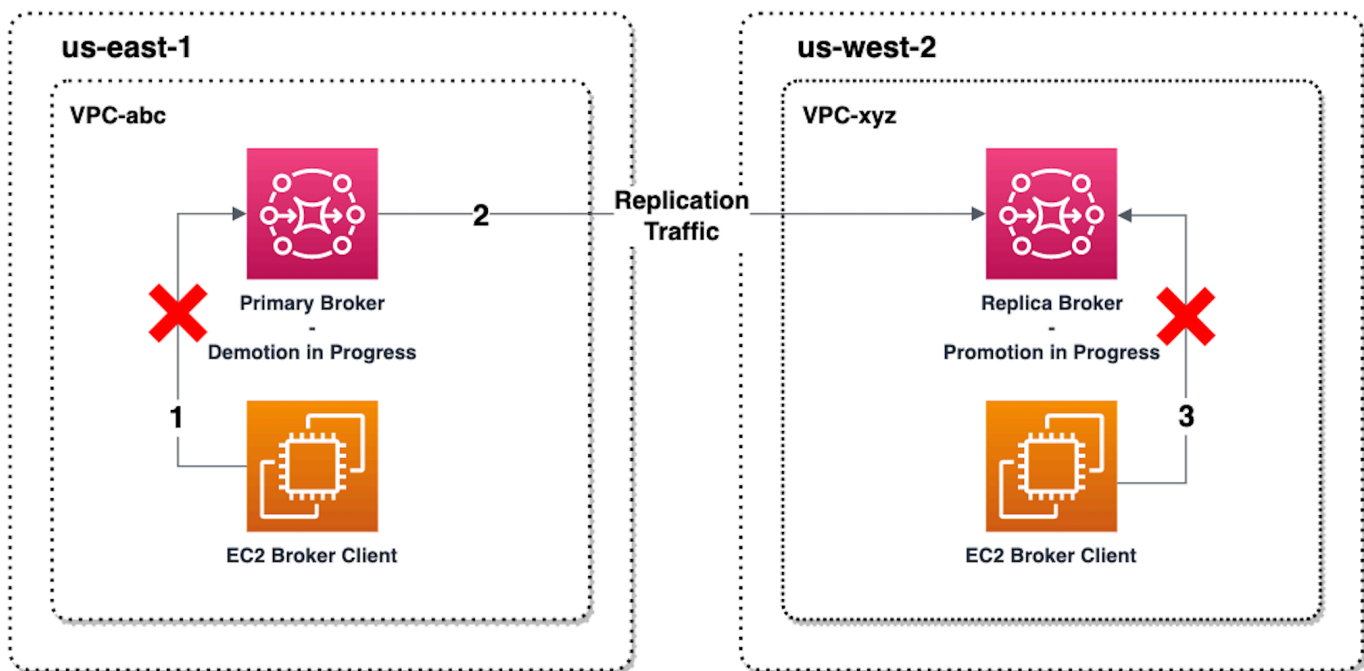
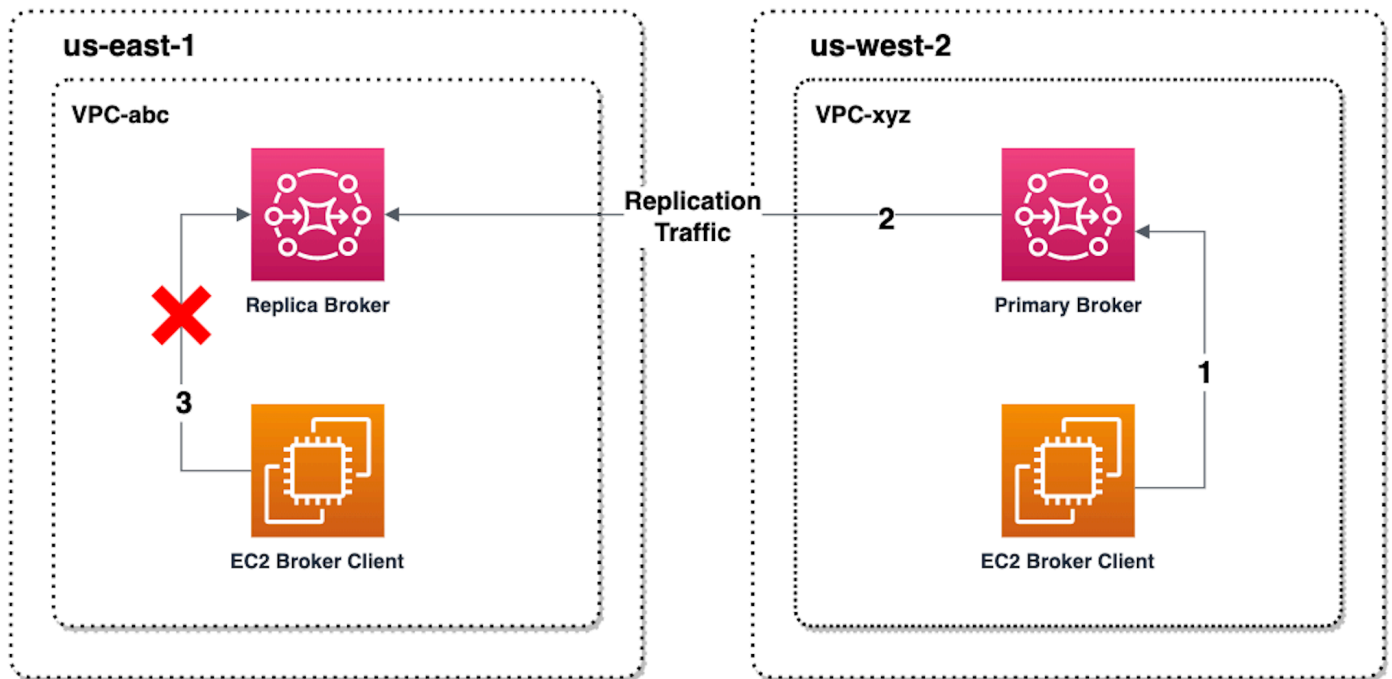


Diagram berikut menggambarkan status broker setelah proses peralihan selesai. Broker replika asli sekarang telah dipromosikan ke peran broker utama dan menerima koneksi klien. Klien dapat memproduksi dan mengonsumsi data dari broker.



## 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, lihat [Mengakses 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.

## Kuota di Amazon MQ untuk ActiveMQ

Topik ini mencantumkan kuota dalam Amazon MQ. Banyak kuota 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](#).

### Note

Untuk kuota di Amazon MQ untuk RabbitMQ, lihat Kuota [di Amazon MQ untuk RabbitMQ](#).

### Topik

- [Pialang](#)
- [Konfigurasi](#)
- [Pengguna](#)
- [Penyimpanan Data](#)
- [Throttling API](#)

## Pialang

Tabel berikut mencantumkan kuota yang terkait dengan Amazon MQ untuk broker ActiveMQ.

Kuota	Deskripsi
Nama broker	<ul style="list-style-type: none"><li>• Harus unik di wilayah broker dan AWS akun Anda.</li><li>• Harus terdiri dari 1-50 karakter.</li></ul>



Kuota	Deskripsi
	<ul style="list-style-type: none"> <li>Hanya dapat berisi karakter yang ditentukan dalam <a href="#">Set Karakter ASCII yang Dapat Dicitak</a>.</li> <li>Hanya dapat berisi karakter alfanumerik, tanda hubung, titik, garis bawah, dan tilde (- . _ ~).</li> </ul>
Jumlah broker, per wilayah	50
Koneksi tingkat kabel per protokol untuk broker yang lebih kecil	300 mq.*.micro misalnya jenis broker.
Koneksi tingkat kabel per protokol untuk broker yang lebih besar	2.000 mq.*.*large misalnya jenis broker.
Jumlah konektor jaringan	20
Grup keamanan per broker	5
Tujuan ActiveMQ (antrian, dan topik) dipantau di CloudWatch	CloudWatch hanya memantau 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 Amazon MQ untuk konfigurasi ActiveMQ.

Kuota	Deskripsi
Nama konfigurasi	<ul style="list-style-type: none"> <li>Harus terdiri dari 1-150 karakter.</li> </ul>

Kuota	Deskripsi
	<ul style="list-style-type: none"> <li>Hanya dapat berisi karakter yang ditentukan dalam <a href="#">Set Karakter ASCII yang Dapat Dicitak</a>.</li> <li>Hanya dapat berisi karakter alfanumerik, tanda hubung, titik, garis bawah, dan tilde (- . _ ~).</li> </ul>
Revisi per konfigurasi	300

## Pengguna

Tabel berikut mencantumkan kuota yang terkait dengan Amazon MQ untuk pengguna broker ActiveMQ.

Kuota	Deskripsi
Nama pengguna	<ul style="list-style-type: none"> <li>Harus terdiri dari 1-100 karakter.</li> <li>Hanya dapat berisi karakter yang ditentukan dalam <a href="#">Set Karakter ASCII yang Dapat Dicitak</a>.</li> <li>Hanya dapat berisi karakter alfanumerik, tanda hubung, titik, garis bawah, dan tilde (- . _ ~).</li> <li>Tidak boleh berisi tanda koma (,).</li> </ul>
Kata Sandi	<ul style="list-style-type: none"> <li>Harus terdiri dari 12-250 karakter.</li> <li>Hanya dapat berisi karakter yang ditentukan dalam <a href="#">Set Karakter ASCII yang Dapat Dicitak</a>.</li> </ul>

Kuota	Deskripsi
	<ul style="list-style-type: none"> <li>• Harus terdiri dari setidaknya 4 karakter unik.</li> <li>• Tidak boleh berisi tanda koma (,).</li> </ul>
Pengguna per broker (auth sederhana)	250
Grup per pengguna (auth sederhana)	20


## Penyimpanan Data

Tabel berikut mencantumkan kuota yang terkait dengan Amazon MQ untuk penyimpanan data ActiveMQ.

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 <a href="#">Broker instance types</a> .
Kapasitas penyimpanan per broker yang lebih besar	200 GB untuk broker tipe instans mq.*.*large. Untuk informasi selengkapnya mengenai tipe instans Amazon MQ, lihat <a href="#">Broker instance types</a> .
Batasan penggunaan penjadwal tugas per broker <a href="#">didukung Amazon EBS</a>	50 GB. Untuk informasi selengkapnya tentang penggunaan penjadwal tugas, lihat <a href="#">JobSchedulerUsage</a> dalam Dokumentasi API Apache ActiveMQ.
Kapasitas penyimpanan sementara per broker yang lebih kecil.	5 GB untuk broker tipe instans mq.*.micro.
Kapasitas penyimpanan sementara per broker yang lebih besar.	50 GB untuk broker tipe instans mq.*.*large.

## Throttling API

Kuota pembatasan berikut dikumpulkan per AWS akun, di semua API Amazon MQ untuk mempertahankan bandwidth layanan. Untuk informasi selengkapnya tentang API Amazon MQ, lihat [Referensi API REST Amazon MQ](#).

 Important

Kuota ini tidak berlaku untuk API olahpesan broker Amazon MQ for ActiveMQ atau Amazon MQ for RabbitMQ. Misalnya, Amazon MQ tidak men-throttle pengiriman atau penerimaan pesan.

Batasan burst API	Batasan laju API
100	15

# Bekerja dengan Amazon MQ for 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 AWS Management Console, REST API Amazon MQ, 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 REST API Amazon MQ, lihat [Referensi REST API Amazon MQ](#).

Topik

- [Mesin RabbitMQ](#)
- [Tutorial RabbitMQ](#)
- [Amazon MQ untuk praktik terbaik RabbitMQ](#)
- [Kuota di Amazon MQ untuk RabbitMQ](#)

## Mesin RabbitMQ

Bagian ini menjelaskan elemen dasar broker RabbitMQ dan plugin yang didukung, serta memberikan gambaran umum tentang opsi arsitektur broker RabbitMQ di Amazon MQ.

Topik

- [Elemen dasar](#)
- [Arsitektur broker](#)
- [Amazon MQ untuk konfigurasi broker RabbitMQ](#)
- [Mengelola Amazon MQ untuk versi mesin RabbitMQ](#)

## Elemen dasar

Bagian ini memperkenalkan konsep-konsep kunci penting untuk memahami RabbitMQ di Amazon MQ.

## Topik

- [Pialang](#)
- [Broker default](#)
- [Jenis instans broker](#)
- [Konfigurasi](#)
- [Pengguna](#)
- [Plugin](#)
- [Kebijakan](#)

## Pialang

Broker adalah lingkungan broker pesan yang berjalan di Amazon MQ. Ini adalah blok bangunan dasar Amazon MQ. Deskripsi gabungan dari instans broker kelas (m5, t3) dan ukuran (large, micro) adalah tipe instans broker (misalnya, mq.m5.large). Untuk informasi selengkapnya, lihat [Broker instance types](#).

- 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, antrian, dan status terdistribusi di beberapa Availability Zone (AZ).

Untuk informasi selengkapnya, lihat [Arsitektur broker](#).

Anda dapat mengaktifkan peningkatan versi minor otomatis ke versi minor baru dari mesin broker, karena versi baru dari mesin RabbitMQ dirilis. 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).

## Protokol yang didukung

Anda dapat mengakses broker RabbitMQ menggunakan [bahasa pemrograman yang didukung RabbitMQ](#) dan dengan mengaktifkan TLS untuk protokol berikut:

- [AMQP \(0-9-1\)](#)

## Port listener

Broker RabbitMQ yang dikelola Amazon MQ mendukung port listener berikut untuk konektivitas tingkat aplikasi melalui amqps, serta koneksi klien menggunakan konsol web RabbitMQ dan API manajemen.

- Port listener 5671 - Digunakan untuk koneksi yang dibuat melalui URL AMQP aman. Sebagai contoh, broker dengan ID broker `b-c8352341-ec91-4a78-ad9c-a43f23d325bb`, di-deploy di wilayah `us-west-2`, berikut adalah URL lengkap amqp broker: `b-c8352341-ec91-4a78-ad9c-a43f23d325bb.mq.us-west-2.amazonaws.com:5671`.
- Port listener 443 dan 15671 - Kedua port listener dapat digunakan secara bergantian untuk mengakses broker melalui konsol web RabbitMQ atau API manajemen.

## Atribut

Broker RabbitMQ memiliki beberapa atribut:

- Nama. Sebagai contoh, `MyBroker`.
- ID. Sebagai contoh, `b-1234a5b6-78cd-901e-2fgh-3i45j6k17819`.
- Amazon Resource Name (ARN). Sebagai contoh, `arn:aws:mq:us-east-2:123456789012:broker:MyBroker:b-1234a5b6-78cd-901e-2fgh-3i45j6k17819`.
- URL konsol web RabbitMQ. Sebagai contoh, `https://b-1234a5b6-78cd-901e-2fgh-3i45j6k17819-1.mq.us-east-2.amazonaws.com`.

Untuk informasi selengkapnya, lihat [konsol web RabbitMQ](#) dalam dokumentasi RabbitMQ.

- Titik akhir aman AMQP. Sebagai contoh, `amqps://b-1234a5b6-78cd-901e-2fgh-3i45j6k17819-1.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](#)

## Broker default

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 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:

- Instans tunggal – `AWS-DEFAULT-POLICY-SINGLE-INSTANCE`
- Deployment klaster – `AWS-DEFAULT-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).

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](#).

- **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

Klaster

- **max-queues:** *number-of-queues-per-vmhost* (batas vmhost) - 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-vmhost* (batas vmhost) - 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

 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
t3.micro	Instans tunggal	N/A	500	500
m5.large	Instans tunggal	N/A	20.000	4.000
	Klaster	8.000.000	4.000	15.000
m5.xlarge	Instans tunggal	N/A	30.000	8.000
	Klaster	9.000.000	5.000	20.000
m5.2xlarge	Instans tunggal	N/A	60.000	15.000
	Klaster	10.000.000	6.000	40.000
m5.4xlarge	Instans tunggal	N/A	150.000	30.000
	Klaster	12.000.000	10.000	100.000

## Jenis instans broker

### Important

Anda tidak dapat menurunkan versi broker dari tipe mq.m5. instans ke tipe mq.t3.micro instans.

### Important

Saat ini Anda tidak dapat membuat t2.micro, m4.large, atau m5.\* broker di zona ketersediaan euc1-az2.

Tipe Instans	vCPU	Memori (GiB)	Kinerja Jaringan	Catatan
mq.t3.micro	2	1	Rendah	<p>Menggunakan tipe instans mq.t3.micro untuk evaluasi dasar Amazon MQ. Tipe instans ini (broker instans tunggal saja) memenuhi syarat untuk <a href="#">Tingkat Gratis AWS</a>.</p> <div style="border: 1px solid #f08080; border-radius: 10px; padding: 10px; margin-top: 10px;"> <p> Important Parameter tipe instans mq.t3.micro tidak mendukung <a href="#">deployment klaster</a>.</p> </div>
mq.m5.large	2	8	Tinggi	Menggunakan instans mq.m5.large untuk pengembangan reguler, pengujian, dan beban kerja produksi.

Tipe Instans	vCPU	Memori (GiB)	Kinerja Jaringan	Catatan
mq.m5.xlarge	4	16	Tinggi	Menggunakan tipe instans mq.m5.xlarge, mq.m5.2xlarge, dan mq.m5.4xlarge untuk pengembangan reguler, pengujian, dan beban kerja produksi yang memerlukan throughput tinggi.
mq.m5.2xlarge	8	32	Tinggi	
mq.m5.4xlarge	16	64	Tinggi	

## Konfigurasi

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

### 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](#). Untuk informasi selengkapnya, lihat [Siklus hidup konfigurasi broker Amazon MQ](#).

Saat ini, Anda tidak dapat menghapus konfigurasi.

Untuk informasi tentang membuat, mengedit, dan mengelola konfigurasi, lihat hal berikut:

- [Creating and applying broker configurations](#)

- [RabbitMQ Broker Configurations](#)

Untuk melacak perubahan yang dibuat pada konfigurasi, Anda dapat membuat revisi konfigurasi. Untuk informasi selengkapnya, lihat [Creating and applying broker configurations](#).

## Atribut

Konfigurasi broker memiliki beberapa atribut, misalnya:

- Nama (`MyConfiguration`)
- ID (`c-1234a5b6-78cd-901e-2fgh-3i45j6k17819`)
- Amazon Resource Name (ARN) (`arn:aws:mq:us-east-2:123456789012:configuration:c-1234a5b6-78cd-901e-2fgh-3i45j6k17819`)

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](#)

## Pengguna

Setiap koneksi klien AMQP 0-9-1 memiliki pengguna terkait yang harus diautentikasi. Setiap koneksi klien juga menargetkan host virtual (vhost) yang mengharuskan pengguna memiliki serangkaian izin. Pengguna mungkin memiliki izin untuk mengonfigurasi, menulis ke, serta membaca dari antrean dan pertukaran di vhost. Kredensial pengguna dan vhost target ditentukan pada saat koneksi dibuat.

Saat pertama kali membuat Amazon MQ untuk broker RabbitMQ, Amazon MQ menggunakan kredensial 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 “tamun”, dan akan menghapus akun tamu default saat Anda membuat broker baru. Amazon MQ juga akan secara berkala menghapus akun yang dibuat pelanggan yang disebut “tamun”.

Untuk membuat pengguna baru dengan API manajemen RabbitMQ, gunakan titik akhir API berikut dan isi permintaan. Ganti *nama pengguna* dan *kata sandi* dengan kredensi 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 lupa kata sandi admin yang Anda tetapkan saat membuat broker, Anda tidak dapat mengatur ulang kredensi Anda. Jika Anda telah membuat beberapa administrator, Anda dapat masuk menggunakan pengguna admin lain dan mengatur ulang atau membuat ulang kredensialnya. Jika Anda hanya memiliki satu pengguna admin, Anda harus menghapus broker dan membuat yang baru dengan kredensi baru. Kami merekomendasikan untuk mengonsumsi atau mencadangkan pesan sebelum menghapus broker.

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 *nama pengguna* dengan informasi Anda. Untuk vhost default /, gunakan %2F.

```
POST /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](#).

## Plugin

Amazon MQ untuk RabbitMQ mendukung [plugin manajemen RabbitMQ yang mendukung API manajemen](#) dan konsol web RabbitMQ. Anda dapat menggunakan konsol web dan API manajemen untuk membuat serta mengelola pengguna dan kebijakan broker.

Selain plugin manajemen, Amazon MQ untuk RabbitMQ juga mendukung plugin berikut.

### Topik

- [Plugin shovel](#)
- [Plugin federasi](#)
- [Plugin pertukaran Hash yang konsisten](#)

### Plugin shovel

Broker terkelola Amazon MQ mendukung [shovel RabbitMQ](#), memungkinkan Anda memindahkan pesan dari antrean dan pertukaran pada satu instans broker ke instans broker lainnya. Anda



dapat menggunakan shovel untuk menghubungkan broker dengan penggabungan longgar dan mendistribusikan pesan dari simpul dengan beban pesan yang lebih berat.

Broker RabbitMQ yang dikelola Amazon MQ mendukung shovel dinamis. Shovel dinamis dikonfigurasi menggunakan parameter waktu aktif, serta 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 shovel dinamis. Dalam contoh, `{vhost}` dapat diganti dengan nama vhost broker, dan `{name}` diganti dengan nama shovel dinamis baru.

```
/api/parameters/shovel/{vhost}/{name}
```

Dalam isi permintaan, Anda harus menentukan antrean atau pertukaran, tidak keduanya. Contoh di bawah ini mengonfigurasi shovel dinamis antara antrean lokal yang ditentukan dalam `src-queue` dan antrean jarak jauh yang ditentukan dalam `dest-queue`. Demikian pula, Anda dapat menggunakan parameter `src-exchange` dan `dest-exchange` untuk mengonfigurasi shovel antara dua pertukaran.

```
{
  "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-
west-2.amazonaws.com:5671",
    "dest-queue": "destination-queue-name"
  }
}
```

#### Important

Anda tidak dapat mengonfigurasi sekop di antara antrian atau pertukaran jika tujuan sekop adalah broker pribadi. Anda hanya dapat mengkonfigurasi sekop antara antrian atau pertukaran di broker publik, atau antara sumber di broker swasta, dan tujuan di broker publik.

Untuk informasi selengkapnya tentang penggunaan shovel dinamis, lihat [Plugin shovel dinamis RabbitMQ](#).

**Note**

Amazon MQ tidak mendukung penggunaan shovel statis.

## Plugin federasi

Amazon MQ mendukung pertukaran dan antrean federasi. 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/my-upstream` dengan isi permintaan berikut.

```
{"value":{"uri":"amqp://server-name","expires":3600000}}
```

- Konfigurasi 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 isi permintaan berikut.

```
{"pattern":"^amq\\.","definition":{"federation-upstream-set":"all"},"apply-to":"exchanges"}
```

### Note

Isi permintaan mengambil pertukaran pada server yang namanya dimulai dengan amq. Menggunakan ekspresi reguler `^amq\\.` akan memastikan bahwa federasi diaktifkan untuk semua pertukaran yang namanya dimulai dengan "amq." Pertukaran di server RabbitMQ Anda dapat diberikan nama yang berbeda.

Untuk informasi selengkapnya tentang mengonfigurasi plugin federasi, lihat [plugin federasi RabbitMQ](#).

## Plugin pertukaran Hash yang konsisten

Secara default, Amazon MQ untuk RabbitMQ mendukung plugin tipe pertukaran Hash Konsisten. 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.

Untuk informasi selengkapnya tentang pertukaran Hash Konsisten, lihat Jenis Pertukaran [Hash Konsisten RabbitMQ di situs](#) web. GitHub

## Kebijakan

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 kebijakan 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


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 (regex).

Tanda	Baca regex	Konfigurasi regex	Tulis regex
administrator	.*	.*	.*

Untuk informasi selengkapnya tentang cara membuat pengguna RabbitMQ serta mengelola tanda dan izin pengguna, lihat [Pengguna](#).


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 antrian pada broker.
- d. Untuk Terapkan ke, pilih Pertukaran dan antrian 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 antrian 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 kluster. `mq.m5.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 diencode sebagai `%2F`. Untuk menerapkan kebijakan ke vhost lain, ganti `%2F` dengan nama vhost.

**Note**

Ganti *nama pengguna* dan *kata sandi* dengan kredensi 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
```

- 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
```

- Untuk membuat batas host virtual `max-connections` yang baru, masukkan perintah `curl` berikut. Perintah ini mengasumsikan antrean pada `vhost / default`, yang diencode sebagai `%2F`. Untuk menerapkan kebijakan ke `vhost` lain, ganti `%2F` dengan nama `vhost`.

**Note**

Ganti *nama pengguna* dan *kata sandi* dengan kredensi masuk administrator Anda. Ganti *number-of-messages* dengan [nilai yang direkomendasikan Amazon MQ](#) sesuai dengan ukuran instans broker dan mode deployment. 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/%2Fmax-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
```

## Arsitektur broker

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

- [Broker instans tunggal](#)
- [Deployment klaster untuk ketersediaan tinggi](#)

## Broker instans tunggal

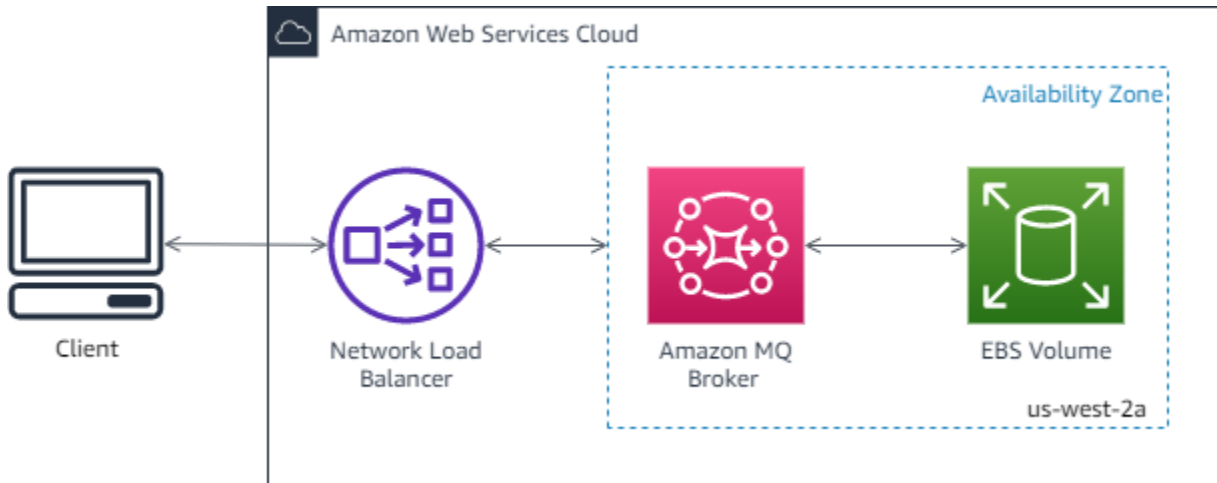
Broker instans tunggal terdiri dari satu broker di satu Availability Zone di balik Penyeimbang Beban Jaringan (NLB). Broker berkomunikasi dengan aplikasi Anda dan dengan volume penyimpanan Amazon EBS. Amazon EBS menyediakan penyimpanan tingkat blok yang dioptimalkan untuk latensi rendah dan throughput tinggi.

Penggunaan Penyeimbang Beban Jaringan memastikan bahwa titik akhir broker Amazon MQ for RabbitMQ tetap tidak berubah jika instans broker diganti selama jendela pemeliharaan atau karena



kegagalan hardware Amazon EC2 yang mendasari. Penyeimbang Beban Jaringan memungkinkan aplikasi dan pengguna Anda untuk terus menggunakan titik akhir yang sama untuk terhubung ke broker.

Diagram berikut mengilustrasikan broker instans tunggal Amazon MQ for RabbitMQ.



## Deployment kluster untuk ketersediaan tinggi

Deployment kluster adalah pengelompokan logis dari tiga node broker RabbitMQ di balik Penyeimbang Beban Jaringan, masing-masing membagikan pengguna, antrian, dan status terdistribusi di beberapa Availability Zone (AZ).

Dalam deployment kluster, 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 antrian memiliki simpul utamanya sendiri. Semua operasi untuk antrian yang diberikan pertama-tama diterapkan pada simpul utama antrian 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 kluster di Availability Zone yang berbeda untuk daya tahan yang lebih baik.


### Note

Selama jendela pemeliharaan, semua pemeliharaan ke kluster 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 kluster

Anda. Untuk informasi selengkapnya tentang pemulihan koneksi, lihat [the section called “Secara otomatis pulih dari kegagalan jaringan”](#).

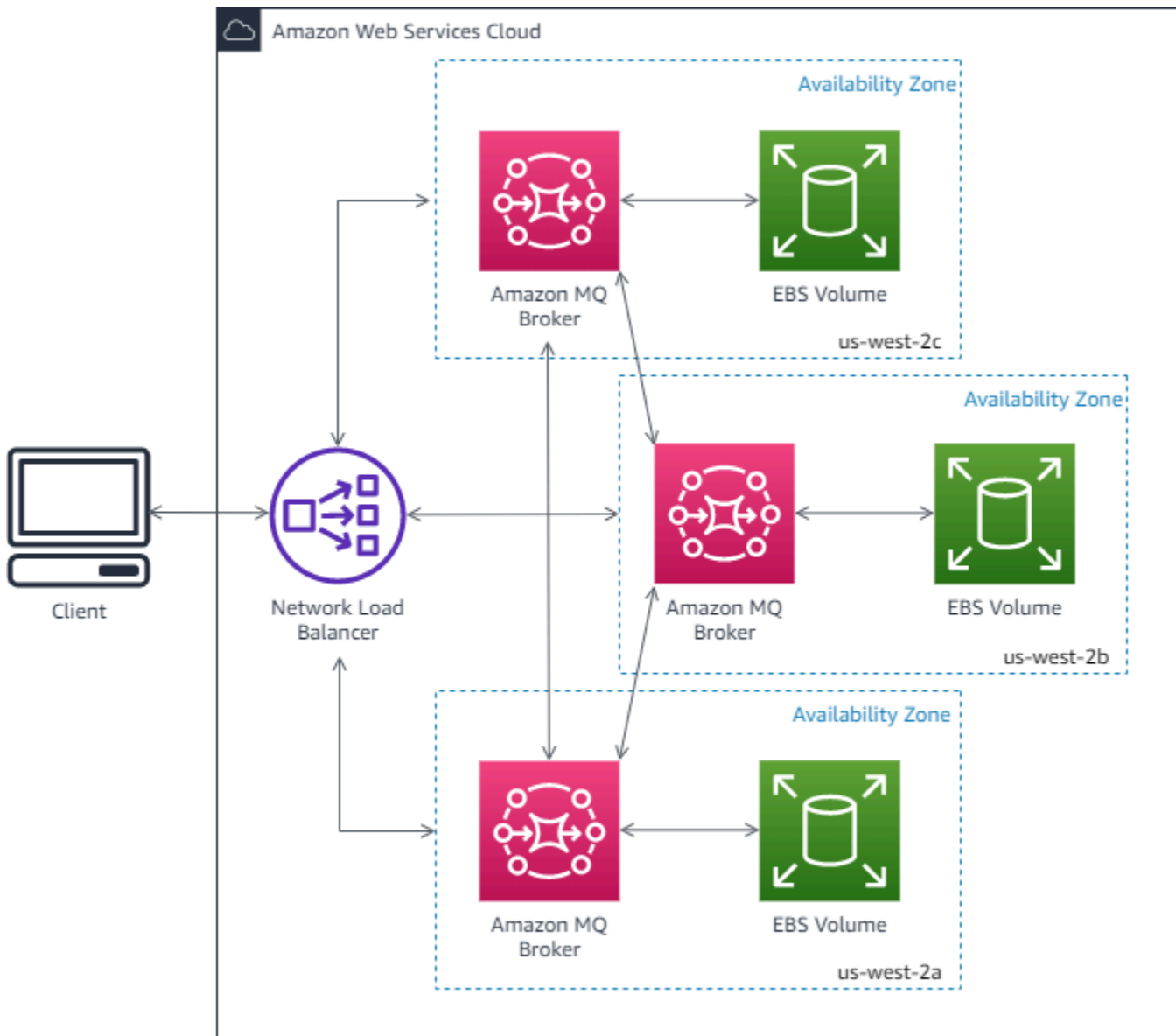
Karena Amazon MQ menetapkan `ha-sync-mode: automatic`, selama jendela pemeliharaan, antrean akan menyinkronkan ketika setiap simpul kembali menggabungkan kluster. Sinkronisasi antrian memblokir semua operasi antrean lainnya. Anda dapat mengurangi dampak sinkronisasi antrean selama jendela pemeliharaan dengan membuat antrian tetap pendek.

Kebijakan default tidak boleh dihapus. Jika Anda menghapus kebijakan ini, Amazon MQ akan membuat lagi kebijakan secara otomatis. Amazon MQ juga akan memastikan bahwa properti HA diterapkan ke semua kebijakan lain yang Anda buat pada broker terkluster. 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](#).

 Important

Amazon MQ tidak mendukung [antrean kuorum](#). Mengaktifkan bendera fitur antrean kuorum dan membuat antrean kuorum akan mengakibatkan kehilangan data.

Diagram berikut mengilustrasikan deployment broker kluster 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 konfigurasi 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.

### Topik

- [Membuat, mengedit, dan menerapkan konfigurasi broker RabbitMQ](#)
- [Kebijakan konfigurasi RabbitMQ](#)

## Membuat, mengedit, 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

Untuk informasi selengkapnya, lihat hal berikut:

- [Konfigurasi](#)
- [Siklus hidup konfigurasi broker Amazon MQ](#)

Contoh berikut menunjukkan bagaimana Anda dapat membuat dan menerapkan konfigurasi broker RabbitMQ menggunakan AWS Management Console

Topik

- [Buat Konfigurasi Baru](#)
- [Buat Revisi Konfigurasi Baru](#)
- [Terapkan Revisi Konfigurasi ke Broker Anda](#)
- [Mengedit Revisi Konfigurasi](#)

Buat Konfigurasi Baru

1. Masuk ke [konsol Amazon MQ](#).
2. Di sebelah kiri, perluas panel navigasi dan pilih Konfigurasi.

**Amazon MQ** ×

Brokers

**Configurations**

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

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](#). Untuk informasi selengkapnya, lihat [Siklus hidup konfigurasi broker Amazon MQ](#).

Saat ini, Anda tidak dapat menghapus konfigurasi.

## Terapkan Revisi Konfigurasi ke Broker Anda


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

**Amazon MQ** ×

**Brokers**

Configurations

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


Broker Anda akan menjadi offline saat sedang di-boot ulang.

5. Pilih Terapkan .

Revisi konfigurasi Anda diterapkan ke broker pada waktu yang ditentukan.

## 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, 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](#). Untuk informasi selengkapnya, lihat [Siklus hidup konfigurasi broker Amazon MQ](#).

Saat ini, Anda tidak dapat menghapus konfigurasi.

## Kebijakan konfigurasi RabbitMQ

Amazon MQ untuk RabbitMQ sekarang mendukung pembuatan dan penerapan konfigurasi ke broker RabbitMQ Anda. 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 AWS Management Console 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. Untuk informasi lebih lanjut tentang menambahkan konfigurasi ke broker Anda, lihat [Creating and applying broker configurations](#).

## 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.Y menunjukkan versi utama, dan mewakili nomor versi minor. Z Amazon MQ menganggap perubahan versi sebagai utama jika nomor versi utama berubah. Misalnya, memutakhirkan dari versi 3.8 hingga 3.9 dianggap sebagai upgrade versi utama. Perubahan versi dianggap minor jika hanya nomor versi minor yang berubah. Misalnya, memutakhirkan dari versi 3.8.23 hingga 3.8.26 dianggap sebagai upgrade versi minor.

Saat Anda membuat Amazon MQ baru untuk broker RabbitMQ, Anda dapat menentukan versi mesin RabbitMQ yang didukung. Jika Anda menggunakan AWS Management Console untuk membuat broker, Amazon MQ secara otomatis default ke nomor versi mesin terbaru. Jika Anda menggunakan AWS CLI atau Amazon MQ API untuk membuat broker, nomor versi mesin diperlukan. Jika Anda tidak memberikan nomor versi, operasi akan menghasilkan pengecualian. Untuk mempelajari selengkapnya, lihat [create-broker](#) di Refensi Perintah AWS CLI dan [CreateBroker](#) dalam Referensi REST API Amazon MQ.

### Important

[Amazon MQ tidak mendukung antrian atau aliran kuorum](#). Mengaktifkan flag fitur ini dan membuat antrian kuorum atau aliran akan mengakibatkan hilangnya data.



### Important

- Amazon MQ tidak mendukung penggunaan logging terstruktur di JSON, diperkenalkan di RabbitMQ 3.9
- Amazon MQ untuk RabbitMQ merekomendasikan broker baru untuk menggunakan versi minor terbaru yang didukung.
- RabbitMQ hanya mengizinkan pembaruan versi tambahan (mis: 3.9.x hingga 3.10.x). Anda tidak dapat melewati versi utama saat memperbarui (mis: 3.8.x ke 3.10.x).

Amazon MQ untuk RabbitMQ saat ini mendukung versi mesin berikut:

Versi utama	Versi minor
RabbitMQ 3.11	<ul style="list-style-type: none"> <li>• 3.11.28 (disarankan)</li> </ul>
RabbitMQ 3.10	<ul style="list-style-type: none"> <li>• 3.10.25</li> </ul>
RabbitMQ 3.9	<ul style="list-style-type: none"> <li>• 3.9.27</li> </ul>
RabbitMQ 3.8	<ul style="list-style-type: none"> <li>• 3.8.34</li> </ul>

Versi minor berikut masih tersedia untuk broker yang ada, tetapi mereka tidak direkomendasikan untuk broker baru.

Versi minor yang ada

Versi utama	Versi minor
RabbitMQ 3.11	<ul style="list-style-type: none"> <li>• 3.11.20</li> <li>• 3.11.16</li> </ul>
RabbitMQ 3.10	<ul style="list-style-type: none"> <li>• 3.10.20</li> <li>• 3.10.10</li> </ul>
RabbitMQ 3.9	<ul style="list-style-type: none"> <li>• 3.9.24</li> <li>• 3.9.16</li> </ul>

Versi utama	Versi minor
	<ul style="list-style-type: none"><li>• 3.9.13</li></ul>
RabbitMQ 3.8	<ul style="list-style-type: none"><li>• 3.8.30</li><li>• 3.8.27</li><li>• 3.8.26</li><li>• 3.8.23</li><li>• 3.8.22</li><li>• 3.8.17</li><li>• 3.8.6</li></ul>

## Pemutakhiran versi utama dan minor

Dengan Amazon MQ, Anda dapat mengontrol waktu untuk meningkatkan broker ke versi baru. Ketika [upgrade versi minor otomatis](#) diaktifkan, Amazon MQ akan secara otomatis meningkatkan mesin broker Anda ke versi minor baru saat dirilis dan didukung oleh Amazon MQ.

Untuk melakukan upgrade versi utama, Anda harus secara manual meng-upgrade nomor versi mesin broker Anda. [Peningkatan versi minor dan mayor terjadi bersamaan dengan operasi patching broker lainnya, selama jendela pemeliharaan terjadwal Anda.](#) Jika Anda memilih keluar dari upgrade versi minor otomatis, Anda dapat secara manual meningkatkan broker Anda ke versi minor baru yang didukung dengan mengikuti prosedur yang sama sebagai upgrade besar.

Untuk informasi lebih lanjut tentang memperbarui preferensi broker Anda untuk mengaktifkan atau menonaktifkan peningkatan versi minor, dan meningkatkan broker Anda secara manual, lihat [the section called “Meningkatkan versi mesin”](#)

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, lihat [Amazon MQ untuk praktik terbaik RabbitMQ](#).

## 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 tipe mq.m5.large instance, ganti tipe mesin dengan dan tipe instance dengan. RABBITMQ mq.m5.large*

## 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 for RabbitMQ](#)
- [Menyelesaikan sinkronisasi antrean RabbitMQ yang dijeda](#)


## Mengedit preferensi broker

Anda dapat mengedit preferensi broker, seperti mengaktifkan atau menonaktifkan CloudWatch Logs menggunakan AWS Management Console.

### Mengedit opsi broker RabbitMQ

1. Masuk ke [konsol Amazon MQ](#).
2. Dari daftar broker, pilih broker Anda (misalnya, MyBroker) lalu pilih Edit.

3. Di halaman Edit **MyBroker**, pada bagian Spesifikasi, pilih Versi mesin broker atau Tipe instans Broker.
4. Di bagian CloudWatch Logs, klik tombol alihkan untuk mengaktifkan atau menonaktifkan log umum. Tidak ada langkah lain yang diperlukan.


 Note

- Untuk broker RabbitMQ, Amazon MQ secara otomatis menggunakan Peran Tertaut Layanan (SLR) untuk memublikasikan 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, konfigurasi jadwal pemeliharaan broker Anda:

Untuk meningkatkan broker ke versi baru saat dirilis AWS, 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 for 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 AMQP 0-9-1 protokol untuk RabbitMQ. Tutorial ini memandu Anda melalui menginstal Pika, mendeklarasikan antrian, menyiapkan penerbit untuk mengirim pesan ke pertukaran default broker, dan menyiapkan konsumen untuk menerima pesan dari antrian.

## Topik

- [Prasyarat](#)
- [Izin](#)
- [Langkah pertama: Buat klien Python Pika dasar](#)
- [Langkah kedua: Membuat penerbit dan mengirimkan pesan](#)
- [Langkah ketiga: Buat konsumen dan terima pesan](#)
- [Langkah keempat: \(Opsional\) Mengatur loop peristiwa dan mengkonsumsi pesan](#)
- [Apa selanjutnya?](#)

## Prasyarat

Untuk menyelesaikan langkah-langkah dalam tutorial ini, Anda memerlukan prasyarat berikut:

- Broker Amazon MQ for RabbitMQ. Untuk informasi selengkapnya, lihat [Membuat broker Amazon MQ untuk RabbitMQ](#).
- [Python 3](#) diinstal untuk sistem operasi Anda.
- [Pika](#) diinstal menggunakan Pythonpip. 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 pengguna broker Amazon MQ for RabbitMQ dengan izin untuk menulis, dan membaca dari, vhost. Tabel berikut menjelaskan izin minimum yang diperlukan sebagai pola ekspresi reguler (regexp).

Tanda	Konfigurasi regexp	Tulis regexp	Baca regexp
none		.*	.*

Izin pengguna yang terdaftar hanya menyediakan izin baca dan tulis kepada pengguna, tanpa memberikan akses ke plugin manajemen untuk melakukan operasi administratif pada broker. Anda selanjutnya dapat membatasi izin dengan menyediakan pola regexp yang membatasi

akses pengguna ke antrian yang ditentukan. Misalnya, jika Anda mengubah pola baca regexp 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 [Pengguna](#).

## Langkah pertama: 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 broker Amazon MQ untuk RabbitMQ, lakukan hal berikut.

1. Buka jendela terminal baru, membuat 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 Anda yang mewarisi dari `BasicPikaClient`.

## Langkah kedua: Membuat penerbit dan mengirimkan pesan

Untuk membuat penerbit yang menyatakan antrian, dan mengirim satu pesan, lakukan hal berikut.

1. Salin konten dari contoh kode berikut, dan simpan secara lokal seperti `publisher.py` dalam direktori yang sama dengan yang Anda buat di 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 delaring 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 dilewatkan 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. Sebagai contoh, `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 sebagai 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 konten dari contoh kode berikut, dan simpan secara lokal seperti `consumer.py` dalam 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 tersebut, ganti parameter yang diteruskan ke `BasicMessageReceiver` konstruktor dengan informasi Anda.
3. Jalankan perintah berikut di direktori proyek Anda.

```
$ python3 consumer.py
```

Jika kode berjalan berhasil, Anda akan melihat isi pesan, dan header termasuk kunci 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 keempat: (Opsional) Mengatur loop peristiwa dan mengonsumsi pesan

Untuk mengonsumsi beberapa pesan dari antrian, gunakan [basic\\_consume](#) metode Pika dan fungsi callback seperti yang ditunjukkan dalam berikut ini

1. Di `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 tetapkan pada 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 yang didukung, lihat [Dokumentasi Klien RabbitMQ](#) di situs RabbitMQ.

## Menyelesaikan sinkronisasi antrean RabbitMQ yang dijeda

Dalam [deployment kluster](#) 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 kluster terdiri dari replika utama di satu simpul dan satu atau lebih cermin. Setiap operasi diterapkan ke antrean cermin, termasuk pesan penambahan antrean, terlebih 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.

### Sinkronisasi pemeliharaan dan antrean

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, menunda sinkronisasi antrean. Anda dapat mengonfirmasi penggunaan memori dengan membandingkan `RabbitMemUsed` dan [metrik `nodeRabbitMqMemLimit` broker di CloudWatch](#). Sinkronisasi tidak dapat diselesaikan hingga pesan dikonsumsi atau dihapus, atau jumlah pesan dalam batch dikurangi.

#### 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	Konfigurasi regexp	Tulis regexp
administrator	.*	.*	.*

Untuk informasi selengkapnya tentang cara membuat pengguna RabbitMQ serta mengelola tanda dan izin pengguna, lihat [Pengguna](#).

## 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. Login ke konsol web RabbitMQ dengan kredensial login 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, kluster Amazon MQ for RabbitMQ dibuat dengan kebijakan broker awal bernama `ha-a11-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.

**ℹ 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 antrian pada `vhost / default`, yang diekode sebagai `%2F`.

**ℹ Note**

Ganti *nama pengguna* dan *kata sandi* dengan kredensi login administrator broker Anda. Anda mungkin perlu menyesuaikan dan mengalibrasi nilai `ha-sync-batch-size` (**100**) berdasarkan jumlah dan ukuran pesan yang tidak disinkronkan dalam antrian. 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-f81n-4ab0-a89c-co62e1c32ef8.mq.us-west-2.amazonaws.com/api/  
policies/%2Fbatch-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-f81n-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 kolom Fitur, antrean Anda harus mencantumkan nama kebijakan baru yang Anda buat (misalnya, `batch-size-policy`).
3. Untuk memulai ulang proses sinkronisasi dengan ukuran batch yang lebih kecil, pilih Mulai ulang sinkronisasi.

### 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 berhasil disinkronkan, Anda dapat memantau jumlah memori yang digunakan simpul RabbitMQ dengan melihat CloudWatch metrik `AmazonRabbitMQMemUsed`. Anda juga dapat melihat metrik `RabbitMQMemLimit` untuk memantau batas memori simpul. Untuk informasi lebih lanjut, lihat [Mengakses CloudWatch metrik untuk Amazon MQ](#) dan [Pencatatan dan pemantauan Amazon MQ untuk broker RabbitMQ](#).
- Agar sinkronisasi antrian tidak dijeda, sebaiknya buat antrian 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 selengkapnya tentang tipe instans broker dan mengedit preferensi broker, lihat [Tipe instans Amazon MQ for RabbitMQ](#) dan [Mengedit 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 [the section called “Broker default”](#).

## Sumber daya terkait

- [UpdateBrokerInput](#)— Gunakan properti broker ini untuk memperbarui tipe instans broker menggunakan API Amazon MQ.
- [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.

## Amazon MQ untuk praktik terbaik RabbitMQ

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

### Important

Amazon MQ tidak mendukung [antrian kuorum](#). Mengaktifkan bendera fitur antrian kuorum dan membuat antrian kuorum akan mengakibatkan kehilangan data.

**⚠ Important**

Saat ini, Amazon MQ tidak mendukung [aliran](#), atau menggunakan logging terstruktur di JSON, diperkenalkan di RabbitMQ 3.9.x.

**⚠ Important**

Amazon MQ untuk RabbitMQ tidak mendukung nama pengguna “tamuh”, dan akan menghapus akun tamu default saat Anda membuat broker baru. Amazon MQ juga akan secara berkala menghapus akun yang dibuat pelanggan yang disebut “tamuh”.

## Topik

- [Mengaktifkan antrean malas](#)
- [Gunakan antrean yang persisten dan tahan lama](#)
- [Jaga antrian pendek](#)
- [Konfigurasi pengakuan dan konfirmasi](#)
- [Konfigurasi pra-pengambilan](#)
- [Konfigurasi Seledri](#)
- [Secara otomatis pulih dari kegagalan jaringan](#)
- [Aktifkan Classic Queue v2 untuk broker RabbitMQ Anda](#)

## Mengaktifkan antrean malas

Jika Anda bekerja dengan antrean yang sangat panjang dan memproses pesan bervolume besar, mengaktifkan antrean malas dapat meningkatkan kinerja keseluruhan broker Anda.

Perilaku default RabbitMQ adalah meng-cache pesan dalam memori dan memindahkannya ke disk hanya ketika broker membutuhkan lebih banyak memori yang tersedia. Proses memindahkan pesan dari memori ke disk dapat memakan waktu dan menghentikan antrean agar tidak memproses pesan. Mengaktifkan antrean malas dapat memiliki dampak yang signifikan pada percepatan proses pemindahan pesan ke disk karena antrean malas menyimpan pesan ke disk sesegera mungkin, sehingga lebih sedikit pesan di-cache dalam memori.

Anda dapat mengaktifkan antrean malas dengan menetapkan argumen `queue.declare` pada saat deklarasi, atau dengan mengonfigurasi kebijakan melalui konsol manajemen RabbitMQ. Contoh berikut menunjukkan cara mendeklarasikan antrean malas menggunakan pustaka klien RabbitMQ Java.

```
Map<String, Object> args = new HashMap<String, Object>();
args.put("x-queue-mode", "lazy");
channel.queueDeclare("myqueue", false, false, false, args);
```

### Note

Mengaktifkan antrean malas dapat meningkatkan operasi I/O disk.

## Gunakan antrean yang persisten dan tahan lama

Pesan persisten dapat membantu mencegah kehilangan data ketika broker lumpuh atau memulai ulang. 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.

```
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());
```

## 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](#).

### Warning

Jangan reboot broker yang menjalankan sumber daya tinggi.

Jika Anda me-reboot broker ketika sinkronisasi antrian dijeda, broker akan memulai kembali proses sinkronisasi, yang selanjutnya dapat menurunkan sumber daya broker karena pesan ditransfer dari penyimpanan ke memori node, dan mengakibatkan broker menjadi tidak responsif di negara bagian. REBOOT\_IN\_PROGRESS

## Konfigurasi pengakuan dan konfirmasi

Ketika aplikasi klien mengirimkan konfirmasi pengiriman dan konsumsi pesan kembali ke broker, ini dikenal sebagai pengakuan konsumen. Demikian pula, proses pengiriman konfirmasi ke penerbit dikenal sebagai konfirmasi penerbit. Pengakuan dan konfirmasi sangat penting untuk memastikan keamanan data saat bekerja dengan broker RabbitMQ.

Pengakuan pengiriman konsumen biasanya dikonfigurasi pada aplikasi klien. Ketika bekerja dengan AMQP 0-9-1, pengakuan dapat diaktifkan dengan mengonfigurasi `basic.consume` atau ketika pesan diambil menggunakan metode `basic.code`.

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.

## Konfigurasi 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 `globaldiatur` ke `false`, sehingga contoh di atas dapat ditulis hanya sebagai `channel.basicQos(10)`.

## Konfigurasi Seledri

Seledri Python mengirimkan banyak pesan yang tidak perlu yang dapat membuat menemukan dan memproses informasi yang berguna menjadi lebih sulit. Untuk mengurangi kebisingan dan mempermudah pemrosesan, masukkan perintah berikut:

```
celery -A app_name worker --without-heartbeat --without-gossip --without-mingle
```

## 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 tidak tertangani dilemparkan ke loop I/O koneksi, jika terdeteksi waktu operasi baca soket habis, atau jika server kehilangan [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.

## Aktifkan Classic Queue v2 untuk broker RabbitMQ Anda

Sebaiknya aktifkan Classic Queue v2 (CQv2) untuk mesin broker versi 3.10 atau lebih tinggi untuk peningkatan kinerja termasuk:

- Ratakan variasi penggunaan memori
- Kurangi penggunaan memori untuk sebagian besar beban kerja
- Tingkatkan pengiriman konsumen
- Meningkatkan throughput untuk beban kerja di mana konsumen mengikuti produsen

Untuk menggunakan CQv2, Anda harus mengaktifkan flag `classic_mirrored_queue_version` fitur terlebih dahulu. Untuk informasi selengkapnya tentang bendera fitur, lihat [Cara mengaktifkan bendera fitur](#).

Untuk bermigrasi dari CQv1 ke CQv2, Anda harus membuat kebijakan antrian baru atau mengedit kebijakan antrian yang ada dengan definisi kunci kebijakan yang disetel ke `queue-version 2`. Untuk informasi selengkapnya tentang penerapan kebijakan, lihat [Kebijakan](#). Untuk informasi selengkapnya tentang mengaktifkan CQv2 dengan kebijakan antrian, lihat Antrian [Klasik dalam dokumentasi RabbitMQ](#).

Migrasi dari CQv1 ke CQv2 dapat menjadi memori intensif untuk antrian di bawah beban tinggi. Sebaiknya ikuti [praktik kinerja terbaik](#) kami yang lain sebelum memulai migrasi.



Jika Anda menggunakan kebijakan antrian, menghapus kebijakan antrian akan menurunkan versi antrian CQv2 kembali ke CQv1. Kami tidak menyarankan untuk menurunkan versi antrian CQv2 ke CQv1 karena RabbitMQ akan mengonversi representasi antrian pada disk. Ini bisa menjadi memori intensif dan memakan waktu untuk antrian dengan kedalaman tinggi.

## Kuota di Amazon MQ untuk RabbitMQ

Topik ini mencantumkan kuota dalam Amazon MQ. Banyak kuota berikut dapat diubah untuk akun AWS spesifik. Untuk meminta kenaikan batas, lihat [AWS Kuota Layanan](#) dan Referensi Umum Amazon Web. Batas yang diperbarui tidak akan terlihat bahkan setelah kenaikan batas telah diterapkan. Untuk informasi lebih lanjut tentang melihat batas koneksi saat ini di AmazonCloudWatch, lihat [Memantau broker Amazon MQ menggunakan AmazonCloudWatch](#).

Topik

- [Pialang](#)
- [Penyimpanan Data](#)
- [Throttling API](#)

## Pialang

Tabel berikut mencantumkan kuota yang terkait dengan Amazon MQ untuk broker RabbitMQ.

Kuota	Deskripsi
Nama broker	<ul style="list-style-type: none"> <li>• Harus unik di wilayah broker dan AndaAWSakun.</li> <li>• Harus terdiri dari 1-50 karakter.</li> <li>• Hanya dapat berisi karakter yang ditentukan dalam <a href="#">Set Karakter ASCII yang Dapat Dicetak</a>.</li> <li>• Hanya dapat berisi karakter alfanumerik, tanda hubung, titik, garis bawah, dan tilde (- . _ ~).</li> </ul>

Kuota	Deskripsi
Jumlah broker, per wilayah	50
Grup keamanan per broker	5
Tujuan ActiveMQ (antrian, dan topik) dipantau diCloudWatch	CloudWatchhanya memonitor 1000 tujuan pertama.
Tujuan RabbitMQ (antrian) dipantau diCloudWatch	CloudWatchmemantau hanya 500 tujuan pertama, dipesan berdasarkan jumlah konsumen.
Tanda per broker	50

## Penyimpanan Data

Tabel berikut mencantumkan kuota yang terkait dengan Amazon MQ untuk penyimpanan data RabbitMQ.

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 <a href="#">Broker instance types</a> .
Kapasitas penyimpanan per broker yang lebih besar	200 GB untuk broker tipe instans mq.*.*large. Untuk informasi selengkapnya mengenai tipe instans Amazon MQ, lihat <a href="#">Broker instance types</a> .

## Throttling API

Kuota throttling berikut diagregat per akun AWS, di semua API Amazon MQ untuk mempertahankan bandwidth layanan. Untuk informasi selengkapnya tentang API Amazon MQ, lihat [Referensi API REST Amazon MQ](#).

**⚠ Important**

Kuota ini tidak berlaku untuk API olahpesan broker Amazon MQ for ActiveMQ atau Amazon MQ for RabbitMQ. Misalnya, Amazon MQ tidak men-throttle pengiriman atau penerimaan pesan.

Batasan burst API	Batasan laju API
100	15

# Keamanan di Amazon MQ

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

Keamanan adalah tanggung jawab bersama antara AWS dan Anda. [Model tanggung jawab bersama](#) menggambarkan ini sebagai keamanan dari cloud dan keamanan di dalam cloud:

- Keamanan cloud – AWS bertanggung jawab untuk melindungi infrastruktur yang menjalankan layanan AWS di Cloud AWS. AWS juga menyediakan layanan yang dapat Anda gunakan dengan aman. Auditor pihak ketiga menguji dan memverifikasi secara berkala efektivitas keamanan kami sebagai bagian dari [AWS Program Kepatuhan](#). Untuk mempelajari tentang program kepatuhan yang berlaku di Amazon MQ, lihat [AWS Layanan dalam Lingkup oleh Program Kepatuhan](#).
- Keamanan dalam cloud – Tanggung jawab Anda ditentukan oleh layanan AWS yang Anda gunakan. Anda juga bertanggung jawab atas faktor lain termasuk sensitivitas data Anda, persyaratan perusahaan Anda, serta hukum dan peraturan yang berlaku.

Dokumentasi ini membantu Anda memahami cara menerapkan model tanggung jawab bersama saat menggunakan Amazon MQ. Topik berikut menunjukkan kepada Anda cara mengonfigurasi Amazon MQ untuk memenuhi tujuan keamanan dan kepatuhan Anda. Anda juga belajar cara menggunakan layanan AWS lainnya yang dapat membantu Anda dalam 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 bersama](#) AWS diterapkan untuk perlindungan data di Amazon MQ.

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

Untuk tujuan perlindungan data, kami menyarankan Anda melindungi Akun AWS kredensial dan mengatur pengguna individu dengan AWS IAM Identity Center atau AWS Identity and Access Management (IAM). Dengan cara tersebut, setiap pengguna hanya diberi izin yang diperlukan untuk memenuhi tugas pekerjaan mereka. Kami juga merekomendasikan agar Anda mengamankan data Anda dengan cara-cara berikut:

- Gunakan autentikasi multi-faktor (MFA) pada setiap akun.
- Gunakan SSL/TLS untuk melakukan komunikasi dengan sumber daya AWS. Kami membutuhkan TLS 1.2 dan merekomendasikan TLS 1.3.
- Siapkan API dan log aktivitas pengguna dengan AWS CloudTrail.
- Gunakan 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 140-2 ketika mengakses AWS melalui antarmuka baris perintah atau API, gunakan titik akhir FIPS. Untuk informasi lebih lanjut tentang titik akhir FIPS yang tersedia, lihat [Standar Pemrosesan Informasi Federal \(FIPS\) 140-2](#).

Kami sangat menyarankan agar Anda tidak pernah memasukkan informasi rahasia atau sensitif, seperti alamat email pelanggan Anda, ke dalam tag atau bidang teks bentuk bebas seperti bidang Nama. Ini termasuk saat Anda bekerja dengan Amazon MQ atau lainnya Layanan AWS menggunakan konsol, API AWS CLI, atau AWS SDK. Data apa pun yang Anda masukkan ke dalam tag atau bidang teks bentuk bebas yang digunakan untuk nama dapat digunakan untuk penagihan atau log diagnostik. Saat Anda memberikan URL ke server eksternal, sebaiknya Anda

tidak menyertakan informasi kredensial di URL untuk memvalidasi permintaan Anda ke server tersebut.

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.

## 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 istirahat

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.
- AWSkunci 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. Sebagai gantinya, kami sarankan untuk menghapus broker jika Anda perlu mencabut hak akses.
- Untuk Amazon MQ untuk broker ActiveMQ yang menggunakan Amazon Elastic File System (EFS) untuk menyimpan data pesan, jika Anda mencabut hibah yang memberikan izin Amazon EFS untuk menggunakan kunci KMS di akun Anda, itu tidak akan segera terjadi.
- 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.
- Menonaktifkan kunci atau mencabut hibah tidak akan segera dilakukan.

Saat membuat [broker instans tunggal](#) dengan kunci KMS untuk RabbitMQ, Anda akan melihat dua `CreateGrant` peristiwa masuk. AWS CloudTrail Acara pertama adalah Amazon MQ yang membuat hibah untuk kunci KMS. Acara kedua adalah EBS membuat hibah untuk EBS untuk digunakan.

CreateGrant AWS CloudTrailentri 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":
      "0ab0acd0b000f00ea00cc0a0e00fc00bce000c000f0000000c0bc0a0000aaafSAMPLE",
      "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 yang 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 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 dalam transit

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 [Mengekripsi Data-AT-Rest dan -In-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\_WITH\_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\_WITH\_AES\_128\_CBC\_SHA256
- TLS\_DHE\_RSA\_WITH\_AES\_128\_CBC\_SHA
- TLS\_RSA\_WITH\_AES\_128\_GCM\_SHA256



- TLS\_RSA\_WITH\_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

- [Penonton](#)
- [Mengautentikasi menggunakan identitas](#)
- [Mengelola akses menggunakan kebijakan](#)
- [Cara kerja Amazon MQ dengan IAM](#)
- [Contoh kebijakan berbasis Identitas Amazon MQ](#)
- [Autentikasi dan otorisasi API untuk Amazon MQ](#)
- [Kebijakan terkelola AWS untuk Amazon MQ](#)
- [Menggunakan peran tertaut layanan untuk Amazon MQ](#)
- [Pemecahan masalah identitas dan akses Amazon MQ](#)

## Penonton

Cara menggunakan AWS Identity and Access Management (IAM) akan berbeda-beda, tergantung pada pekerjaan yang Anda lakukan di Amazon MQ.

**Pengguna layanan** – Jika Anda menggunakan layanan Amazon MQ untuk melakukan tugas Anda, administrator Anda akan memberikan kredensial dan izin yang dibutuhkan. Saat Anda menggunakan lebih banyak fitur Amazon MQ untuk melakukan tugas, Anda mungkin memerlukan izin tambahan. Memahami bagaimana cara mengelola akses dapat membantu Anda meminta izin yang tepat dari administrator Anda. Jika Anda tidak dapat mengakses fitur di Amazon MQ, lihat [Pemecahan masalah identitas dan akses Amazon MQ](#).

**Administrator layanan** – Jika Anda bertanggung jawab atas sumber daya Amazon MQ di korporasi Anda, Anda mungkin memiliki akses penuh ke Amazon MQ. Tugas Anda adalah menentukan fitur dan sumber daya Amazon MQ mana yang harus diakses pengguna layanan Anda. Anda kemudian harus mengirimkan permintaan ke administrator IAM Anda untuk mengubah izin pengguna layanan Anda. Tinjau informasi di halaman ini untuk memahami konsep Basic IAM. Pelajari selengkapnya tentang cara korporasi Anda dapat menggunakan IAM dengan Amazon MQ, lihat [Cara kerja Amazon MQ dengan IAM](#).

**Administrator IAM** – Jika Anda adalah administrator IAM, Anda mungkin ingin belajar dengan lebih terpedetail tentang cara Anda dapat menulis kebijakan untuk mengelola akses ke Amazon MQ. Untuk melihat kebijakan contoh berbasis identitas Amazon MQ yang dapat Anda gunakan di IAM, lihat [Contoh kebijakan berbasis Identitas Amazon MQ](#).

## Mengautentikasi menggunakan identitas

Autentikasi adalah cara Anda untuk masuk ke AWS menggunakan kredensial identitas Anda. Anda harus diautentikasi (masuk ke AWS) sebagai Pengguna root akun AWS, sebagai pengguna IAM, atau dengan mengasumsikan peran IAM.

Anda dapat masuk AWS sebagai identitas federasi dengan menggunakan kredensial yang disediakan melalui sumber identitas. AWS IAM Identity Center Pengguna (IAM Identity Center), autentikasi masuk tunggal perusahaan Anda, dan kredensial Google atau Facebook Anda adalah contoh identitas federasi. Saat Anda masuk sebagai identitas federasi, administrator Anda sebelumnya menyiapkan federasi identitas menggunakan peran IAM. Ketika Anda mengakses AWS dengan menggunakan federasi, Anda secara tidak langsung mengambil peran.

Bergantung pada jenis pengguna Anda, Anda dapat masuk ke AWS Management Console atau portal AWS akses. Untuk informasi selengkapnya tentang masuk AWS, lihat [Cara masuk ke Panduan AWS Sign-In Pengguna Anda Akun AWS](#).

Jika Anda mengakses AWS secara terprogram, AWS sediakan kit pengembangan perangkat lunak (SDK) dan antarmuka baris perintah (CLI) untuk menandatangani permintaan Anda secara kriptografis dengan menggunakan kredensial Anda. Jika Anda tidak menggunakan AWS alat, Anda harus menandatangani permintaan sendiri. Untuk informasi selengkapnya tentang penggunaan metode yang disarankan untuk menandatangani permintaan sendiri, lihat [Menandatangani permintaan AWS API](#) di Panduan Pengguna IAM.

Terlepas dari metode otentikasi yang Anda gunakan, Anda mungkin diminta untuk memberikan informasi keamanan tambahan. Misalnya, AWS menyarankan supaya Anda menggunakan autentikasi multi-faktor (MFA) untuk meningkatkan keamanan akun Anda. Untuk mempelajari lebih lanjut, lihat [Autentikasi multi-faktor di Panduan AWS IAM Identity Center Pengguna dan Menggunakan otentikasi multi-faktor \(MFA\) AWS](#) di Panduan Pengguna IAM.

## Pengguna root Akun AWS

Saat Anda membuat Akun AWS, Anda mulai dengan satu identitas masuk yang memiliki akses lengkap ke semua Layanan AWS dan sumber daya di akun. Identitas ini disebut pengguna root Akun AWS dan diakses dengan cara masuk menggunakan alamat email dan kata sandi yang Anda gunakan saat membuat akun. Kami sangat menyarankan agar Anda tidak menggunakan pengguna root untuk tugas sehari-hari Anda. Lindungi kredensial pengguna root Anda dan gunakan untuk melakukan tugas-tugas yang hanya dapat dilakukan oleh pengguna root. Untuk daftar lengkap tugas yang mengharuskan Anda masuk sebagai pengguna root, lihat [Tugas yang memerlukan kredensial pengguna root di Panduan Pengguna IAM](#).

## Pengguna dan grup

[Pengguna IAM](#) adalah identitas dalam akun Akun AWS Anda yang memiliki izin khusus untuk satu orang atau aplikasi. Jika memungkinkan, sebaiknya mengandalkan kredensial sementara daripada membuat pengguna IAM yang memiliki kredensial jangka panjang seperti kata sandi dan kunci akses. Namun, jika Anda memiliki kasus penggunaan khusus yang memerlukan kredensial jangka panjang dengan pengguna IAM, kami sarankan Anda memutar kunci akses. Untuk informasi selengkapnya, lihat [Memutar kunci akses secara teratur untuk kasus penggunaan yang memerlukan kredensial jangka panjang](#) di Panduan Pengguna IAM.

[Grup IAM](#) adalah identitas yang menentukan kumpulan dari para pengguna IAM. Anda tidak dapat masuk sebagai kelompok. Anda dapat menggunakan grup untuk menentukan izin untuk beberapa pengguna sekaligus. Grup membuat izin lebih mudah dikelola untuk set besar pengguna. Misalnya, Anda dapat memiliki grup yang diberi nama IAMAdmins dan memberikan izin kepada grup tersebut untuk mengelola sumber daya IAM.

Para pengguna berbeda dari peran. Seorang pengguna secara unik terkait dengan satu orang atau aplikasi, tetapi peran ini dimaksudkan untuk dapat digunakan oleh siapa pun yang membutuhkannya. Pengguna memiliki kredensial jangka panjang permanen, tetapi peran memberikan kredensial sementara. Untuk mempelajari selengkapnya, lihat [Kapan harus membuat pengguna IAM \(bukan peran\)](#) dalam Panduan Pengguna IAM.

## IAM role

[IAM role](#) adalah identitas dalam akun Akun AWS Anda yang memiliki izin khusus. Peran ini mirip dengan pengguna IAM, tetapi tidak terkait dengan orang tertentu. Anda dapat menggunakan IAM role untuk sementara di dalam AWS Management Console dengan cara [berganti peran](#). Anda dapat mengambil peran dengan cara memanggil operasi API AWS CLI atau AWS atau menggunakan URL khusus. Untuk informasi selengkapnya tentang metode penggunaan peran, lihat [Menggunakan IAM roles](#) dalam Panduan Pengguna IAM.

IAM role dengan kredensial sementara berguna dalam situasi berikut:

- Akses pengguna terfederasi — Untuk menetapkan izin ke identitas federasi, Anda membuat peran dan menentukan izin untuk peran tersebut. Ketika identitas federasi mengautentikasi, identitas dikaitkan dengan peran dan diberikan izin yang ditentukan oleh peran. Untuk informasi tentang peran untuk federasi, lihat [Membuat peran untuk Penyedia Identitas pihak ketiga](#) di Panduan Pengguna IAM. Jika Anda menggunakan Pusat Identitas IAM, Anda mengonfigurasi set izin. Untuk mengontrol apa yang dapat diakses identitas Anda setelah diautentikasi, IAM Identity Center mengkorelasikan izin yang disetel ke peran dalam IAM. Untuk informasi tentang set izin, lihat [Set izin](#) di Panduan AWS IAM Identity Center Pengguna.
- Izin pengguna IAM sementara — Pengguna atau peran IAM dapat mengambil peran IAM untuk sementara mengambil izin yang berbeda untuk tugas tertentu.
- Akses lintas akun – Anda dapat menggunakan IAM role agar seseorang (principal tepercaya) di akun lain diizinkan untuk mengakses sumber daya di akun Anda. Peran adalah cara utama untuk memberikan akses lintas akun. Namun, dengan beberapa Layanan AWS, Anda dapat melampirkan kebijakan secara langsung ke sumber daya (alih-alih menggunakan peran sebagai proxy). Untuk mempelajari perbedaan antara kebijakan peran dan kebijakan berbasis sumber daya untuk akses

lintas akun, lihat [Perbedaan antara IAM role dan kebijakan berbasis sumber daya](#) dalam Panduan Pengguna IAM.

- Akses lintas layanan — Beberapa Layanan AWS menggunakan fitur lain Layanan AWS. Sebagai contoh, ketika Anda melakukan panggilan dalam suatu layanan, biasanya layanan tersebut menjalankan aplikasi di Amazon EC2 atau menyimpan objek di Amazon S3. Layanan mungkin melakukan ini menggunakan izin panggilan principal, menggunakan peran layanan, atau peran terkait layanan.
- Sesi akses teruskan (FAS) — Saat Anda menggunakan pengguna atau peran IAM untuk melakukan tindakan AWS, Anda dianggap sebagai prinsipal. Ketika Anda menggunakan beberapa layanan, Anda mungkin melakukan tindakan yang kemudian memulai tindakan lain di layanan yang berbeda. FAS menggunakan izin dari pemanggilan utama Layanan AWS, dikombinasikan dengan permintaan Layanan AWS untuk membuat permintaan ke layanan hilir. Permintaan FAS hanya dibuat ketika layanan menerima permintaan yang memerlukan interaksi dengan orang lain Layanan AWS atau sumber daya untuk menyelesaikannya. Dalam hal ini, Anda harus memiliki izin untuk melakukan kedua tindakan tersebut. Untuk detail kebijakan saat membuat permintaan FAS, lihat [Meneruskan sesi akses](#).
- Peran layanan – Peran layanan adalah [IAM role](#) yang diambil oleh layanan untuk melakukan tindakan atas nama Anda. Administrator IAM dapat membuat, memodifikasi, dan menghapus peran layanan dari dalam IAM. Untuk informasi selengkapnya, lihat [Membuat peran untuk mendelegasikan izin ke Layanan AWS dalam Panduan Pengguna IAM](#).
- Peran terkait layanan — Peran terkait layanan adalah jenis peran layanan yang ditautkan ke Layanan AWS. Layanan dapat menggunakan peran untuk melakukan tindakan atas nama Anda. Peran terkait layanan muncul di Akun AWS dan dimiliki oleh layanan. Administrator IAM dapat melihat, tetapi tidak dapat mengedit izin untuk peran terkait layanan.
- Aplikasi yang berjalan di Amazon EC2 – Anda dapat menggunakan IAM role untuk mengelola kredensial sementara untuk aplikasi yang berjalan pada instans EC2, dan membuat permintaan API AWS CLI atau AWS. Menyimpan access key di dalam instans EC2 lebih disarankan. Untuk menugaskan sebuah peran AWS ke instans EC2 dan membuatnya tersedia untuk semua aplikasinya, Anda dapat membuat sebuah profil instans yang dilampirkan ke instans. Profil instans berisi peran dan memungkinkan program yang berjalan di instans EC2 untuk mendapatkan kredensial sementara. Untuk informasi selengkapnya, lihat [Menggunakan IAM role untuk memberikan izin ke aplikasi yang berjalan di instans Amazon EC2](#) dalam Panduan Pengguna IAM.

Untuk mempelajari kapan waktunya menggunakan IAM role atau pengguna IAM, lihat [Kapan harus membuat IAM role \(bukan pengguna\)](#) dalam Panduan Pengguna IAM.

## Mengelola akses menggunakan kebijakan

Anda mengontrol akses AWS dengan membuat kebijakan dan melampirkannya ke AWS identitas atau sumber daya. Kebijakan adalah objek AWS yang, ketika dikaitkan dengan identitas atau sumber daya, menentukan izinnya. AWS mengevaluasi kebijakan ini ketika prinsipal (pengguna, pengguna root, atau sesi peran) membuat permintaan. Izin dalam kebijakan dapat menentukan permintaan yang diizinkan atau ditolak. Sebagian besar kebijakan disimpan di AWS sebagai dokumen JSON. Untuk informasi selengkapnya tentang struktur dan isi dokumen kebijakan JSON, lihat [Gambaran Umum kebijakan JSON](#) dalam Panduan Pengguna IAM.

Administrator dapat menggunakan kebijakan JSON AWS untuk menentukan siapa yang memiliki akses ke hal apa. Yaitu, principal mana yang dapat melakukan tindakan pada sumber daya apa, dan dalam syarat apa.

Secara default, pengguna dan peran tidak memiliki izin. Untuk memberikan izin kepada pengguna untuk melakukan tindakan pada sumber daya yang mereka butuhkan, administrator IAM dapat membuat kebijakan IAM. Administrator kemudian dapat menambahkan kebijakan IAM ke peran, dan pengguna dapat mengambil peran.

Kebijakan IAM mendefinisikan izin untuk suatu tindakan terlepas dari metode yang Anda gunakan untuk pengoperasiannya. Misalnya, Anda memiliki kebijakan yang mengizinkan tindakan `iam:GetRole`. Pengguna dengan kebijakan tersebut dapat memperoleh informasi peran dari API AWS Management Console, the AWS CLI, or the AWS.

### Kebijakan berbasis identitas

Kebijakan berbasis identitas adalah dokumen kebijakan izin JSON yang dapat Anda lampirkan ke identitas, seperti pengguna IAM, grup pengguna, atau peran. Kebijakan ini mengontrol tipe tindakan yang dapat dilakukan oleh pengguna dan peran, di sumber daya mana, dan dalam syarat. Untuk mempelajari cara membuat kebijakan berbasis identitas, lihat [Membuat kebijakan IAM](#) dalam Panduan Pengguna IAM.

Kebijakan berbasis identitas dapat dikategorikan lebih lanjut sebagai kebijakan inline atau kebijakan terkelola. Kebijakan inline disematkan secara langsung ke satu pengguna, grup, atau peran. Kebijakan terkelola adalah kebijakan mandiri yang dapat Anda lampirkan ke beberapa pengguna, grup, dan peran dalam akun AWS Anda. Kebijakan terkelola meliputi kebijakan yang dikelola AWS dan kebijakan yang dikelola pelanggan. Untuk mempelajari cara memilih antara kebijakan terkelola atau kebijakan inline, lihat [Memilih 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. Contoh kebijakan berbasis sumber daya adalah kebijakan tepercaya IAM role dan kebijakan bucket Amazon S3. Dalam layanan yang mendukung kebijakan berbasis sumber daya, administrator layanan dapat menggunakannya untuk mengontrol akses ke sumber daya tertentu. Untuk sumber daya yang dilampiri kebijakan tersebut, kebijakan menetapkan tindakan apa yang dapat dilakukan oleh principal tertentu di sumber daya tersebut dan dalam kondisi apa. Anda harus [menentukan principal](#) dalam kebijakan berbasis sumber daya. Prinsipal dapat mencakup akun, pengguna, peran, pengguna federasi, atau. Layanan AWS

Kebijakan berbasis sumber daya merupakan kebijakan inline yang terletak di layanan tersebut. Anda tidak dapat menggunakan kebijakan yang dikelola AWS dari IAM dalam kebijakan berbasis sumber daya.

## Daftar Kontrol Akses (ACL)

Daftar kontrol akses (ACL) mengendalikan principal mana (anggota akun, pengguna, atau peran) yang memiliki izin untuk mengakses sumber daya. ACL serupa dengan kebijakan berbasis sumber daya, meskipun tidak menggunakan format dokumen kebijakan JSON.

Amazon S3, AWS WAF, dan Amazon VPC adalah contoh layanan yang mendukung ACL. Untuk mempelajari ACL selengkapnya, lihat [Gambaran umum daftar kontrol akses \(ACL\)](#) dalam Panduan Developer Amazon Simple Storage Service.

## Tipe kebijakan lainnya

AWS mendukung tipe kebijakan tambahan, yang kurang umum. Tipe kebijakan ini dapat mengatur izin maksimum yang diberikan kepada Anda oleh tipe kebijakan yang lebih umum.

- Batasan izin – Batasan izin adalah fitur lanjutan tempat Anda mengatur izin maksimum yang dapat diberikan oleh kebijakan berbasis identitas kepada entitas IAM (pengguna atau IAM role). Anda dapat menetapkan batas izin untuk suatu entitas. Izin yang dihasilkan adalah persimpangan kebijakan berbasis identitas entitas dan batas izinnnya. Kebijakan berbasis sumber daya yang menentukan pengguna atau peran dalam bidang Principal tidak dibatasi oleh batasan izin. Penolakan eksplisit dalam salah satu kebijakan ini dapat membatalkan izin. Untuk informasi selengkapnya tentang batasan izin, lihat [Batasan izin untuk entitas IAM](#) dalam Panduan Pengguna IAM.



- Kebijakan Kontrol Layanan (SCPs) – SCP adalah kebijakan JSON yang menentukan izin maksimum untuk sebuah organisasi atau unit organisasi (OU) di AWS Organizations. AWS Organizations adalah layanan untuk mengelompokkan dan mengelola beberapa akun AWS secara terpusat yang dimiliki oleh bisnis Anda. Jika Anda mengaktifkan semua fitur di sebuah organisasi, maka Anda dapat menerapkan kebijakan kontrol layanan (SCP) ke salah satu atau ke semua akun Anda. SCP membatasi izin untuk entitas dalam akun anggota, termasuk setiap Pengguna root akun AWS. Untuk informasi selengkapnya tentang Organizations dan SCP, lihat [Cara kerja SCP](#) dalam Panduan Pengguna AWS Organizations.
- Kebijakan sesi – Kebijakan sesi adalah kebijakan lanjutan yang Anda berikan sebagai parameter saat Anda membuat sesi sementara secara terprogram bagi peran atau pengguna gabungan. Izin sesi yang dihasilkan adalah persimpangan kebijakan berbasis identitas pengguna atau peran dan kebijakan sesi. Izin juga dapat berasal dari kebijakan berbasis sumber daya. Penolakan eksplisit dalam salah satu kebijakan ini membatalkan izin. Untuk informasi selengkapnya, lihat [Kebijakan sesi](#) dalam Panduan Pengguna IAM.

## Berbagai tipe kebijakan

Ketika beberapa tipe kebijakan berlaku untuk sebuah permintaan, izin yang dihasilkan lebih rumit untuk dipahami. Untuk mempelajari cara AWS menentukan untuk mengizinkan permintaan ketika beberapa tipe kebijakan dilibatkan, lihat [Logika evaluasi kebijakan](#) dalam 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 Amazon MQ dan layanan AWS lainnya yang bekerja dengan IAM, lihat [Layanan AWS yang bekerja dengan IAM](#) dalam Panduan Pengguna IAM.

Amazon MQ menggunakan IAM untuk membuat, memperbarui, dan menghapus operasi, tapi autentikasi ActiveMQ native untuk broker. Untuk informasi selengkapnya, lihat [Mengintegrasikan broker ActiveMQ dengan LDAP](#).

### 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 JSON AWS untuk menentukan siapa yang memiliki akses ke hal apa. Yaitu, principal mana yang dapat melakukan tindakan pada sumber daya apa, dan dalam syarat apa.

Elemen `Action` dari kebijakan JSON menjelaskan tindakan-tindakan yang dapat Anda gunakan untuk mengizinkan atau menolak akses dalam sebuah kebijakan. Tindakan kebijakan biasanya memiliki nama yang sama sebagai operasi API AWS terkait. Ada beberapa pengecualian, misalnya tindakan hanya dengan izin yang tidak memiliki operasi API yang cocok. Ada juga beberapa operasi yang memerlukan beberapa tindakan dalam kebijakan. Tindakan tambahan ini disebut tindakan dependen.

Sertakan tindakan dalam kebijakan untuk memberikan izin guna melakukan operasi 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 JSON AWS untuk menentukan siapa yang memiliki akses ke hal apa. Yaitu, principal mana yang dapat melakukan tindakan pada sumber daya apa, dan dalam syarat apa.

Elemen kebijakan JSON Resource menentukan objek atau objek-objek yang menjadi target penerapan tindakan. Pernyataan harus mencakup elemen Resource atau NotResource. Sebagai praktik terbaik, tentukan sumber daya menggunakan [Amazon Resource Name \(ARN\)](#). Anda dapat melakukan ini untuk tindakan yang mendukung tipe sumber daya tertentu, yang dikenal sebagai izin tingkat sumber daya.

Untuk tindakan yang tidak mendukung izin tingkat sumber daya, seperti operasi daftar, gunakan wildcard (\*) untuk menunjukkan bahwa pernyataan berlaku untuk semua sumber daya.

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

Dalam Amazon MQ, sumber daya primer AWS adalah broker pesan Amazon MQ dan konfigurasinya. Setiap broker dan konfigurasi Amazon MQ memiliki Amazon Resource Name (ARN) yang unik dan terkait, seperti yang ditunjukkan pada tabel berikut.

Jenis Sumber Daya	ARN	Kunci kondisi
brokers	arn:aws:mq:us-east-1:123456789012:broker:\${brokerName}:\${brokerId}	<a href="#">aws:ResourceTag/\${TagKey}</a>
configurations	arn:\${Partition}:mq:\${Region}:\${Account}:configuration:\${configuration-id}	<a href="#">aws:ResourceTag/\${TagKey}</a>

Untuk informasi lebih lanjut tentang format ARN, lihat [Amazon Resource Name \(ARN\) dan namespace layanan AWS](#).

Misalnya, untuk menentukan broker bernama `MyBroker` dengan `BrokerIDb-1234a5b6-78cd-901e-2fgh-3i45j6k17819` dalam pernyataan Anda, gunakan ARN berikut:

```
"Resource": "arn:aws:mq:us-east-1:123456789012:broker:MyBroker:b-1234a5b6-78cd-901e-2fgh-3i45j6k17819"
```

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 pernyataan tunggal, pisahkan ARN dengan koma.

```
"Resource": [  
  "resource1",  
  "resource2"
```

Untuk melihat daftar tipe sumber daya Amazon MQ dan ARN mereka, lihat [Sumber Daya Ditetapkan 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 kondisi

Administrator dapat menggunakan kebijakan JSON AWS untuk menentukan siapa yang memiliki akses ke hal apa. Yaitu, prinsipal mana yang dapat melakukan tindakan pada sumber daya apa, dan menurut persyaratan apa.

Elemen `Condition` (atau `Condition` blok) memungkinkan Anda menentukan syarat di mana suatu pernyataan berlaku. Elemen `Condition` bersifat opsional. Anda dapat membuat ekspresi bersyarat

yang menggunakan [operator syarat](#), seperti sama dengan atau kurang dari, untuk mencocokkan syarat dalam kebijakan dengan nilai dalam permintaan.

Jika Anda menentukan beberapa elemen `Condition` dalam pernyataan, atau beberapa kunci dalam satu elemen `Condition`, AWS akan mengevaluasinya dengan menggunakan operasi logika AND. Jika Anda menetapkan beberapa nilai untuk kunci syarat tunggal, AWS akan mengevaluasi syarat tersebut dengan menggunakan operasi logika OR. Semua persyaratan harus dipenuhi sebelum izin pernyataan diberikan.

Anda juga dapat menggunakan variabel placeholder saat menentukan syarat. Sebagai contoh, Anda dapat memberikan izin pengguna IAM untuk mengakses sumber daya hanya jika ditandai dengan nama pengguna IAM mereka. Untuk informasi lebih lanjut, lihat [Elemen kebijakan IAM: variabel dan tag](#) dalam Panduan Pengguna IAM.

AWS mendukung kunci syarat global dan kunci syarat khusus layanan. Untuk melihat semua kunci syarat global AWS, lihat [AWS kunci konteks syarat global](#) dalam 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
<a href="#">aws:RequestTag/\${TagKey}</a>	Filter tindakan berdasarkan tanda yang diberikan dalam permintaan.	String
<a href="#">aws:ResourceTag/\${TagKey}</a>	Filter tindakan berdasarkan tanda yang terkait dengan sumber daya.	String
<a href="#">aws:TagKeys</a>	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`:

```
{
  "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:

- [Penandaan sumber daya](#)
- [Mengontrol Akses Menggunakan Tag IAM](#)

## Peran IAM Amazon MQ

[IAM role](#) adalah entitas dalam akun AWS Anda yang memiliki izin khusus.

Menggunakan kredensial sementara dengan Amazon MQ

Anda dapat menggunakan kredensial sementara untuk masuk dengan federasi, memiliki IAM role, atau menjalankan peran lintas-akun. Anda mendapatkan kredensial keamanan sementara dengan menelepon AWS STS Operasi API seperti [AssumeRole](#) atau [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 API AWS Management Console, AWS CLI, or AWS. Administrator IAM harus membuat kebijakan IAM yang memberikan izin kepada pengguna dan peran untuk melakukan operasi API tertentu pada sumber daya yang diperlukan. Administrator kemudian harus melampirkan kebijakan tersebut ke pengguna 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 terkena biaya. Ketika Anda membuat atau mengedit kebijakan berbasis identitas, ikuti panduan dan rekomendasi ini:

- Memulai kebijakan AWS terkelola dan beralih ke izin paling sedikit hak istimewa — Untuk mulai memberikan izin kepada pengguna dan beban kerja Anda, gunakan kebijakan AWS terkelola yang memberikan izin untuk banyak kasus penggunaan umum. Mereka tersedia di Akun AWS. Kami menyarankan Anda mengurangi izin lebih lanjut dengan menentukan kebijakan yang dikelola AWS pelanggan yang spesifik untuk kasus penggunaan Anda. Untuk informasi selengkapnya, lihat [kebijakan AWS terkelola atau kebijakan terkelola untuk fungsi pekerjaan](#) di Panduan Pengguna IAM.
- Berikan izin paling rendah — Saat Anda menetapkan izin dengan kebijakan IAM, hanya berikan izin yang diperlukan untuk melaksanakan tugas. Anda melakukan ini dengan menentukan tindakan yang dapat diambil pada sumber daya tertentu dalam kondisi tertentu, juga dikenal sebagai izin paling tidak memiliki hak istimewa. Untuk informasi selengkapnya tentang penggunaan IAM untuk menerapkan izin, lihat [Kebijakan dan izin di IAM](#) dalam Panduan Pengguna IAM.
- Gunakan ketentuan dalam kebijakan IAM untuk membatasi akses lebih lanjut — Anda dapat menambahkan kondisi pada kebijakan Anda untuk membatasi akses ke tindakan dan sumber daya. Misalnya, Anda dapat menulis kebijakan untuk menentukan bahwa semua permintaan harus dikirim menggunakan SSL. Anda juga dapat menggunakan kondisi untuk memberikan akses ke tindakan layanan jika digunakan melalui spesifik Layanan AWS, seperti AWS CloudFormation. Untuk informasi lebih lanjut, lihat [Elemen kebijakan IAM JSON: Syarat](#) dalam Panduan Pengguna IAM.
- Gunakan IAM Access Analyzer untuk memvalidasi kebijakan IAM Anda untuk memastikan izin yang aman dan fungsional - IAM Access Analyzer memvalidasi kebijakan baru dan yang sudah ada sehingga kebijakan mematuhi bahasa kebijakan IAM (JSON) dan praktik terbaik IAM. IAM Access Analyzer menyediakan lebih dari 100 pemeriksaan kebijakan dan rekomendasi yang dapat ditindaklanjuti untuk membantu Anda membuat kebijakan yang aman dan fungsional. Untuk informasi selengkapnya, lihat [validasi kebijakan IAM Access Analyzer](#) di Panduan Pengguna IAM.
- Memerlukan otentikasi multi-faktor (MFA) — Jika Anda memiliki skenario yang mengharuskan pengguna IAM atau pengguna root di Akun AWS, aktifkan MFA untuk keamanan tambahan.

Untuk mewajibkan MFA saat operasi API dipanggil, tambahkan kondisi MFA ke kebijakan Anda. Untuk informasi selengkapnya, lihat [Mengonfigurasi akses API yang dilindungi MFA](#) di Panduan Pengguna IAM.

Untuk informasi selengkapnya tentang praktik terbaik dalam IAM, lihat [praktik terbaik keamanan di IAM](#) dalam Panduan Pengguna IAM.

## Menggunakan konsol Amazon MQ

Untuk mengakses konsol Amazon MQ, Anda harus memiliki satu set izin minimum. Izin ini harus memperbolehkan Anda untuk membuat daftar dan melihat detail tentang sumber daya Amazon MQ di akun AWS Anda. Jika Anda membuat kebijakan berbasis identitas yang lebih ketat dari izin minimum yang diperlukan, konsol tidak akan berfungsi sebagaimana dimaksudkan untuk entitas (pengguna atau IAM role) dengan kebijakan tersebut.

Untuk memastikan bahwa entitas tersebut masih dapat menggunakan konsol Amazon MQ, lampirkan kebijakan terkelola AWS berikut ini ke entitas. Untuk informasi selengkapnya, lihat [Menambahkan Izin ke Pengguna](#) dalam Panduan Pengguna IAM.

```
AmazonMQReadOnlyAccess
```

Anda tidak perlu mengizinkan konsol minimum untuk pengguna yang melakukan panggilan hanya ke AWS CLI atau API AWS. Alih-alih, izinkan akses hanya ke tindakan yang sesuai dengan operasi API yang Anda coba lakukan.

## Izinkan para pengguna untuk melihat izin mereka sendiri

Contoh ini menunjukkan cara Anda dapat membuat kebijakan yang mengizinkan para pengguna IAM untuk melihat kebijakan inline dan terkelola yang dilampirkan ke identitas pengguna mereka. Kebijakan ini mencakup izin untuk menyelesaikan tindakan pada konsol atau secara terprogram menggunakan API AWS CLI atau AWS.

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Sid": "ViewOwnUserInfo",
      "Effect": "Allow",
      "Action": [
        "iam:GetUserPolicy",
```



```

        "iam:ListGroupsWithUser",
        "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 permintaan AWS standar untuk autentikasi 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 mengotorisasi pengguna AWS agar dapat bekerja dengan broker, konfigurasi, dan pengguna, Anda harus mengedit izin kebijakan IAM.

### Topik

- [Izin IAM yang Diperlukan untuk Membuat Broker Amazon MQ](#)

- [Referensi izin REST API 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.

```
{
  "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 daftar REST API Amazon MQ dan izin IAM yang sesuai.

### REST API Amazon MQ dan Izin yang Diperlukan

REST API Amazon MQ	Izin yang Diperlukan
<a href="#">CreateBroker</a>	mq:CreateBroker
<a href="#">CreateConfiguration</a>	mq:CreateConfiguration
<a href="#">CreateTags</a>	mq:CreateTags
<a href="#">CreateUser</a>	mq:CreateUser
<a href="#">DeleteBroker</a>	mq>DeleteBroker
<a href="#">DeleteUser</a>	mq>DeleteUser
<a href="#">DescribeBroker</a>	mq:DescribeBroker
<a href="#">DescribeConfiguration</a>	mq:DescribeConfiguration
<a href="#">DescribeConfigurationRevision</a>	mq:DescribeConfigurationRevision

REST API Amazon MQ	Izin yang Diperlukan
<a href="#">DescribeUser</a>	mq:DescribeUser
<a href="#">ListBrokers</a>	mq:ListBrokers
<a href="#">ListConfigurationRevisions</a>	mq:ListConfigurationRevisions
<a href="#">ListConfigurations</a>	mq:ListConfigurations
<a href="#">ListTags</a>	mq:ListTags
<a href="#">ListUsers</a>	mq:ListUsers
<a href="#">RebootBroker</a>	mq:RebootBroker
<a href="#">UpdateBroker</a>	mq:UpdateBroker
<a href="#">UpdateConfiguration</a>	mq:UpdateConfiguration
<a href="#">UpdateUser</a>	mq:UpdateUser

## 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 API Amazon MQ yang saat ini mendukung izin tingkat sumber daya, serta sumber daya yang didukung, ARN sumber daya, dan kunci syarat untuk setiap tindakan.

### 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)
<a href="#">CreateConfiguration</a>	<a href="#">konfigurasi</a> *
<a href="#">CreateTags</a>	<a href="#">broker</a> , <a href="#">konfigurasi</a>
<a href="#">CreateUser</a>	<a href="#">pialang</a> *
<a href="#">DeleteBroker</a>	<a href="#">pialang</a> *
<a href="#">DeleteUser</a>	<a href="#">pialang</a> *
<a href="#">DescribeBroker</a>	<a href="#">pialang</a> *
<a href="#">DescribeConfiguration</a>	<a href="#">konfigurasi</a> *
<a href="#">DescribeConfigurationRevision</a>	<a href="#">konfigurasi</a> *
<a href="#">DescribeUser</a>	<a href="#">pialang</a> *
<a href="#">ListConfigurationRevisions</a>	<a href="#">konfigurasi</a> *
<a href="#">ListConfigurationRevisions</a>	<a href="#">konfigurasi</a> *
<a href="#">ListTags</a>	<a href="#">broker</a> , <a href="#">konfigurasi</a>
<a href="#">ListUsers</a>	<a href="#">pialang</a> *
<a href="#">RebootBroker</a>	<a href="#">pialang</a> *
<a href="#">UpdateBroker</a>	<a href="#">pialang</a> *
<a href="#">UpdateConfiguration</a>	<a href="#">konfigurasi</a> *
<a href="#">UpdateUser</a>	<a href="#">pialang</a> *

## Kebijakan terkelola AWS untuk Amazon MQ

Sebuah AWS kebijakan terkelola adalah kebijakan mandiri yang dibuat dan dikelola oleh AWS. AWS kebijakan terkelola dirancang untuk memberikan izin untuk banyak kasus penggunaan umum sehingga Anda dapat mulai menetapkan izin kepada pengguna, grup, dan peran.

Perlu diingat bahwa AWS kebijakan terkelola mungkin tidak memberikan izin hak istimewa untuk kasus penggunaan spesifik Anda karena tersedia untuk semua AWS pelanggan untuk digunakan. Kami menyarankan Anda mengurangi izin lebih lanjut dengan mendefinisikan [kebijakan yang dikelola pelanggan](#) yang khusus untuk kasus penggunaan Anda.

Anda tidak dapat mengubah izin yang ditentukan dalam AWS kebijakan yang dikelola. Jika AWS memperbarui izin yang didefinisikan dalam AWS kebijakan terkelola, pembaruan mempengaruhi semua identitas utama (pengguna, grup, dan peran) yang dilampirkan kebijakan. AWS kemungkinan besar akan memperbarui AWS kebijakan 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.

### AWS kebijakan terkelola: AmazonMQServiceRolePolicy

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 untuk kebijakan terkelola AWS

Lihat detail tentang pembaruan ke kebijakan terkelola AWS untuk Amazon MQ karena 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 terkelola AWS.	5 Mei 2021

## Menggunakan peran tertaut layanan untuk Amazon MQ

Amazon MQ menggunakan [peran tertaut layanan](#) AWS Identity and Access Management (IAM). Peran tertaut layanan adalah tipe IAM role unik yang ditautkan langsung ke Amazon MQ. Peran tertaut layanan telah ditentukan sebelumnya oleh Amazon MQ dan mencakup semua izin yang diperlukan layanan untuk memanggil layanan AWS lainnya atas nama Anda.

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 `AWSServiceRoleForAmazonMQ` – Amazon MQ menggunakan peran terkait layanan ini untuk memanggil layanan AWS atas nama Anda.

Peran terkait layanan `AWSServiceRoleForAmazonMQ` memercayai layanan berikut untuk menjalankan peran:

- `mq.amazonaws.com`

Amazon MQ menggunakan kebijakan izin [AmazonMQServiceRolePolicy](#), yang terlampir ke peran terkait layanan `AWSServiceRoleForAmazonMQ` 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.
- Membuat grup log, dan memublikasikan log broker ke Amazon CloudWatch Logs.



```
{
  "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, menyunting, atau menghapus peran terhubung dengan 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. Ketika Anda pertama kali membuat broker, Amazon MQ membuat peran terkait layanan untuk memanggil layanan AWS 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 API AWS, buat peran yang terhubung dengan layanan dengan nama layanan `mq.amazonaws.com`. Untuk informasi lebih lanjut, lihat [Membuat peran terkait layanan](#) dalam Panduan Pengguna IAM. Jika Anda menghapus peran yang terhubung dengan layanan ini, Anda dapat menggunakan proses yang sama untuk membuat ulang peran tersebut.

## Mengedit peran tertaut layanan untuk Amazon MQ

Amazon MQ tidak mengizinkan Anda untuk mengedit peran tertaut layanan `AWSServiceRoleForAmazonMQ`. Namun, Anda dapat menyunting 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.

**ℹ 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 lagi.

Untuk menghapus sumber daya Amazon MQ yang digunakan oleh `AWSServiceRoleForAmazonMQ`

- Menghapus broker Amazon MQ menggunakan AWS Management Console, CLI Amazon MQ, atau API Amazon MQ. Untuk informasi lebih lanjut tentang penghapusan broker, lihat [???](#).

Untuk menghapus peran terkait layanan secara manual menggunakan IAM

Gunakan konsol IAM, AWS CLI, atau API AWS untuk menghapus peran terkait layanan `AWSServiceRoleForAmazonMQ`. 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
Asia Pacific (Sydney)	ap-southeast-2	Ya
Asia Pacific (Tokyo)	ap-northeast-1	Ya
Canada (Central)	ca-central-1	Ya
Eropa (Frankfurt)	eu-central-1	Ya

Nama wilayah	Identitas wilayah	Dukungan di Amazon MQ
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 diotorisasi untuk melakukan iam:M:PassRole](#)
- [Saya ingin mengizinkan orang di luar akun AWS saya untuk mengakses sumber daya Amazon MQ saya](#)

### Saya tidak Berwenang untuk Melakukan Tindakan di Amazon MQ

Jika AWS Management Console memberi tahu bahwa Anda tidak berwenang untuk melakukan tindakan, Anda harus menghubungi administrator untuk mendapatkan bantuan. Administrator Anda adalah orang yang memberikan kredensial masuk Anda.

Contoh kesalahan berikut terjadi ketika `mateojackson` pengguna mencoba menggunakan konsol untuk melihat detail tentang suatu `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 agar dia dapat mengakses sumber daya `my-example-widget` menggunakan tindakan `mq:GetWidget`.

## Saya tidak diotorisasi untuk melakukan iam:M:PassRole

Jika Anda menerima pesan kesalahan bahwa Anda tidak terotorisasi untuk melakukan `iam:PassRole` tindakan, kebijakan Anda harus diperbarui agar Anda memberikan peran ke Amazon MQ.

Beberapa Layanan AWS memungkinkan Anda memberikan peran yang sudah ada ke layanan tersebut alih-alih membuat peran layanan baru atau peran tertaut 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 mengharuskan layanan untuk memiliki izin yang diberikan oleh peran layanan. Mary tidak memiliki izin untuk meneruskan peran tersebut ke layanan.

```
User: arn:aws:iam::123456789012:user/marymajor is not authorized to perform:
iam:PassRole
```

Dalam kasus ini, kebijakan Mary harus diperbarui agar dia melakukan `iam:PassRole` tindakan tersebut.

Jika Anda memerlukan bantuan, hubungi AWS administrator Anda. Administrator Anda adalah orang yang memberikan kredensial masuk Anda.

## Saya ingin mengizinkan orang di luar akun AWS saya untuk mengakses sumber daya Amazon MQ saya

Anda dapat membuat peran yang dapat digunakan pengguna di akun atau orang lain di luar organisasi Anda untuk mengakses sumber daya Anda. Anda dapat menentukan siapa yang dipercaya untuk mengambil peran tersebut. Untuk layanan yang mendukung kebijakan berbasis sumber daya atau daftar kontrol akses (ACL), Anda dapat menggunakan kebijakan tersebut untuk memberi akses pada orang ke sumber daya Anda.

Untuk mempelajari selengkapnya, lihat hal berikut:

- Untuk mempelajari apakah Amazon MQ mendukung fitur ini, lihat [Cara kerja Amazon MQ dengan IAM](#).
- Untuk mempelajari cara memberikan akses ke sumber daya di seluruh Akun AWS yang Anda miliki, lihat [Menyediakan akses ke pengguna IAM di akun Akun AWS lain yang Anda miliki](#) dalam Panduan Pengguna IAM.

- Untuk mempelajari cara memberikan akses ke sumber daya Anda ke Akun AWS pihak ketiga, lihat [Menyediakan akses ke akun Akun AWS yang dimiliki oleh pihak ketiga](#) dalam Panduan Pengguna IAM.
- Untuk mempelajari cara memberikan akses melalui federasi identitas, lihat [Menyediakan akses ke pengguna terautentikasi eksternal \(gabungan identitas\)](#) dalam Panduan Pengguna IAM .
- Untuk mempelajari perbedaan antara penggunaan peran dan kebijakan berbasis sumber daya untuk akses lintas akun, lihat [Perbedaan IAM role dan kebijakan berbasis sumber daya](#) di Panduan Pengguna IAM.

## Validasi kepatuhan untuk Amazon MQ

Auditor pihak ke tiga menilai keamanan dan kepatuhan Amazon MQ sebagai bagian dari beberapa program kepatuhan AWS. 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 Kepatuhan AWS](#).

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

Tanggung jawab kepatuhan Anda saat menggunakan Layanan AWS ditentukan oleh sensitivitas data Anda, tujuan kepatuhan perusahaan Anda, serta hukum dan peraturan yang berlaku. AWS menyediakan sumber daya berikut untuk membantu kepatuhan:

- [Panduan Memulai Cepat Keamanan dan Kepatuhan — Panduan](#) penerapan ini membahas pertimbangan arsitektur dan memberikan langkah-langkah untuk menerapkan lingkungan dasar AWS yang berfokus pada keamanan dan kepatuhan.
- [Arsitektur untuk Keamanan dan Kepatuhan HIPAA di Amazon Web Services](#) — Whitepaper ini menjelaskan bagaimana perusahaan dapat menggunakan AWS untuk membuat aplikasi yang memenuhi syarat HIPAA.

### Note

Tidak semua Layanan AWS memenuhi syarat HIPAA. Untuk informasi selengkapnya, lihat [Referensi Layanan yang Memenuhi Syarat HIPAA](#).

- [Sumber Daya Kepatuhan AWS](#) – Kumpulan buku kerja dan panduan ini mungkin berlaku untuk industri dan lokasi Anda.
- [AWS Panduan Kepatuhan Pelanggan](#) - Memahami model tanggung jawab bersama melalui lensa kepatuhan. Panduan ini merangkum praktik terbaik untuk mengamankan Layanan AWS dan memetakan panduan untuk kontrol keamanan di berbagai kerangka kerja (termasuk Institut Standar dan Teknologi Nasional (NIST), Dewan Standar Keamanan Industri Kartu Pembayaran (PCI), dan Organisasi Internasional untuk Standardisasi (ISO)).
- [Mengevaluasi Sumber Daya dengan Aturan](#) di Panduan Developer AWS Config – Layanan AWS Config menilai seberapa baik konfigurasi sumber daya Anda dalam mematuhi praktik-praktik internal, pedoman industri, dan regulasi internal.
- [AWS Security Hub](#)— Ini Layanan AWS memberikan pandangan komprehensif tentang keadaan keamanan Anda di dalamnya AWS. Security Hub menggunakan kontrol keamanan untuk mengevaluasi AWS sumber daya Anda dan untuk memeriksa kepatuhan Anda terhadap standar industri keamanan dan praktik terbaik. Untuk daftar layanan dan kontrol yang didukung, lihat [referensi kontrol Security Hub](#).
- [AWS Audit Manager](#) – Layanan AWS ini akan membantu Anda untuk terus-menerus mengaudit penggunaan AWS untuk menyederhanakan bagaimana Anda mengelola risiko dan kepatuhan terhadap regulasi dan standar industri.

## Ketahanan di Amazon MQ

Infrastruktur global AWS dibangun di sekitar Wilayah AWS dan Availability Zones. AWS Wilayah menyediakan beberapa Availability Zone yang terpisah secara fisik dan terisolasi, yang terhubung dengan jaringan berlatensi rendah, throughput yang tinggi, dan sangat redundan. Dengan Availability Zone, Anda dapat mendesain dan mengoperasikan aplikasi dan basis data yang secara otomatis mengalami kegagalan di antara zona tanpa gangguan. Availability Zone lebih tersedia, memiliki toleransi kesalahan, dan dapat diskalakan dibandingkan dengan satu atau beberapa infrastruktur pusat data tradisional.

Untuk informasi selengkapnya tentang Wilayah AWS dan Availability Zone, lihat [Infrastruktur Global AWS](#).

## Keamanan infrastruktur di Amazon MQ

Sebagai layanan terkelola, dilindungi oleh AWS keamanan jaringan global. Untuk informasi tentang AWS layanan keamanan dan bagaimana AWS melindungi infrastruktur, lihat [AWS Keamanan](#)



[Cloud](#). Untuk mendesain AWS lingkungan menggunakan praktik terbaik untuk keamanan infrastruktur, lihat [Perlindungan Infrastruktur](#) di Pilar Keamanan AWS Kerangka Kerja yang Diarsiteksikan.

Anda menggunakan AWS panggilan API yang dipublikasikan untuk mengakses melalui jaringan. Klien harus mendukung hal berikut:

- Transport Layer Security (TLS). Kami membutuhkan TLS 1.2 dan merekomendasikan TLS 1.3.
- Suite cipher dengan kerahasiaan maju sempurna (PFS) seperti DHE (Ephemeral Diffie-Hellman) atau ECDHE (Elliptic Curve Ephemeral Diffie-Hellman). Sebagian besar sistem modern seperti Java 7 dan sistem yang lebih baru mendukung mode ini.

Selain itu, permintaan harus ditandatangani menggunakan access key ID dan secret access key yang terkait dengan principal IAM. Atau Anda bisa menggunakan [AWS Security Token Service](#) (AWS STS) untuk membuat kredensial keamanan sementara guna menandatangani permintaan.

## Praktik terbaik keamanan untuk Amazon MQ

Pola desain berikut dapat meningkatkan keamanan broker Amazon MQ Anda.

Topik

- [Lebih memilih broker tanpa aksesibilitas publik](#)
- [Selalu konfigurasi 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. Hal ini sangat mengurangi kerentanan broker terhadap serangan Distributed Denial of Service (DDoS) dari internet publik. Untuk informasi selengkapnya, lihat [Mengakses konsol web broker tanpa aksesibilitas publik](#) dalam panduan ini dan [How to Help Prepare for DDoS Attacks by Reducing Your Attack Surface](#) dalam Blog Keamanan AWS.

## Selalu konfigurasi 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, Anda harus membatasi koneksi protokol dan port yang tidak diperlukan melalui konfigurasi Grup Keamanan Amazon VPC yang benar. Misalnya, untuk membatasi akses ke sebagian besar protokol seraya memungkinkan akses ke OpenWire dan konsol web, Anda dapat mengizinkan akses ke 61617 dan 8162 saja. Ini membatasi paparan Anda dengan memblokir protokol yang tidak digunakan, seraya memungkinkan OpenWire dan konsol web berfungsi secara normal.

Hanya memungkinkan port protokol yang Anda gunakan.

- AMQP: 5671
- MQTT: 8883
- OpenWire: 61617
- MENGINJAK-INJAK: 61614
- WebSocket: 61619

Untuk informasi selengkapnya, lihat:

- [Configure Additional Broker Settings](#)
- [Grup Keamanan untuk VPC Anda](#)
- [Grup Keamanan Default untuk VPC Anda](#)

- [Bekerja dengan Grup Keamanan](#)

# Pencatatan dan pemantauan broker Amazon MQ

Pemantauan adalah bagian penting dari pemeliharaan keandalan, ketersediaan, dan performa solusi AWS Anda. Anda harus mengumpulkan data pemantauan dari semua bagian solusi AWS agar dapat dengan lebih mudah melakukan debug kegagalan multi-titik jika terjadi. AWS menyediakan beberapa alat untuk memantau sumber daya Amazon MQ Anda dan merespons potensi insiden:

## Topik

- [Mengakses CloudWatch metrik untuk Amazon MQ](#)
- [Memantau broker Amazon MQ menggunakan Amazon CloudWatch](#)
- [Pencatatan panggilan API Amazon MQ menggunakan AWS CloudTrail](#)
- [Mengonfigurasi Amazon MQ untuk memublikasikan log ke Amazon CloudWatch Logs](#)

## Mengakses CloudWatch metrik untuk Amazon MQ

Amazon MQ dan Amazon CloudWatch terintegrasi sehingga Anda dapat menggunakan CloudWatch untuk melihat dan menganalisis metrik untuk broker ActiveMQ Anda dan tujuan broker (antrian dan topik). Anda dapat melihat dan menganalisis metrik Amazon MQ Anda dari CloudWatch konsol, AWS CLI, atau CLI. CloudWatch CloudWatchmetrik untuk Amazon MQ secara otomatis disurvei dari broker dan kemudian didorong ke CloudWatch setiap menit.

Untuk daftar lengkap metrik Amazon MQ, lihat [Monitoring Amazon MQ using CloudWatch](#).

Untuk informasi tentang membuat CloudWatch alarm untuk metrik, lihat [Membuat atau Mengedit CloudWatch Alarm](#) di Panduan CloudWatch Pengguna Amazon.

### Note

Tidak ada biaya untuk metrik Amazon MQ yang dilaporkan. CloudWatch Metrik ini disediakan sebagai bagian dari layanan Amazon MQ.

Untuk broker ActiveMQ, CloudWatch monitor hanya 1000 tujuan pertama.

Untuk broker RabbitMQ, CloudWatch monitor hanya 500 tujuan pertama, dipesan oleh jumlah konsumen.

## Topik

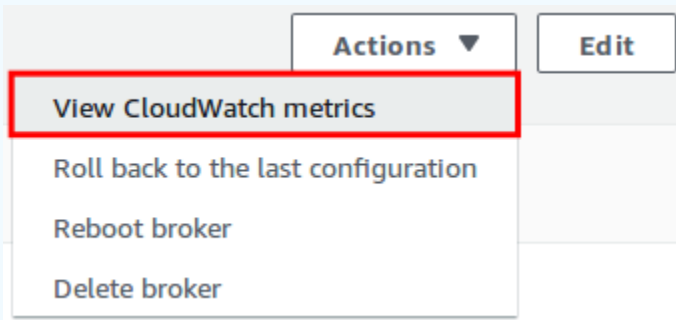
- [AWS Management Console](#)
- [AWS Command Line Interface](#)
- [CloudWatchAPI Amazon](#)

## AWS Management Console

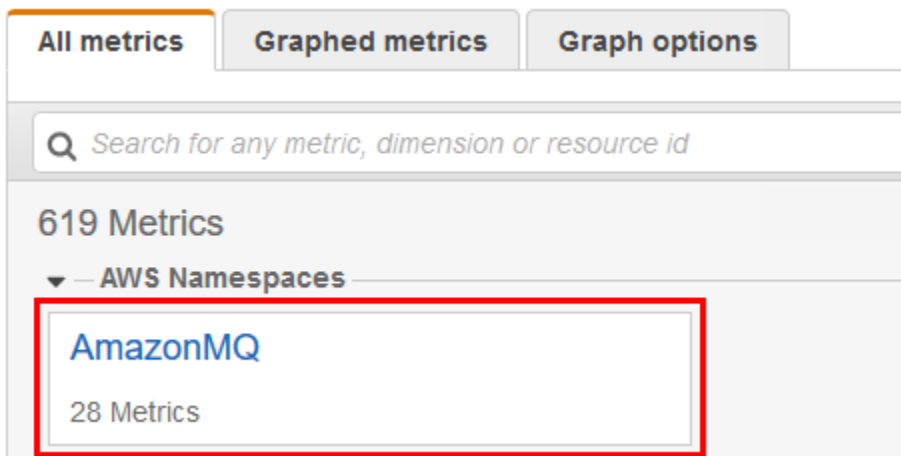
Contoh berikut menunjukkan cara mengakses CloudWatch metrik untuk Amazon MQ menggunakan AWS Management Console

### Note

Jika Anda sudah masuk ke konsol Amazon MQ, di halaman Detail broker, pilih Tindakan, Lihat CloudWatch metrik.



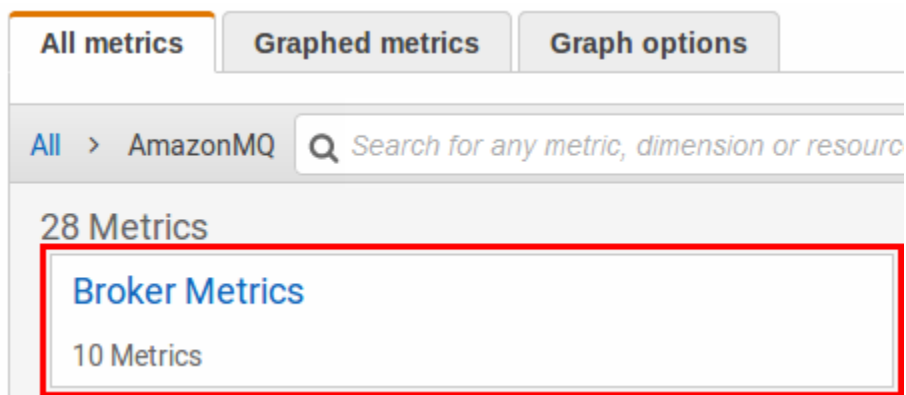
1. Masuk ke [konsol CloudWatch](#) tersebut.
2. Di panel navigasi, pilih Metrics (Metrik).
3. Pilih namespace metrik AmazonMQ.



#### 4. Pilih salah satu dimensi metrik berikut:

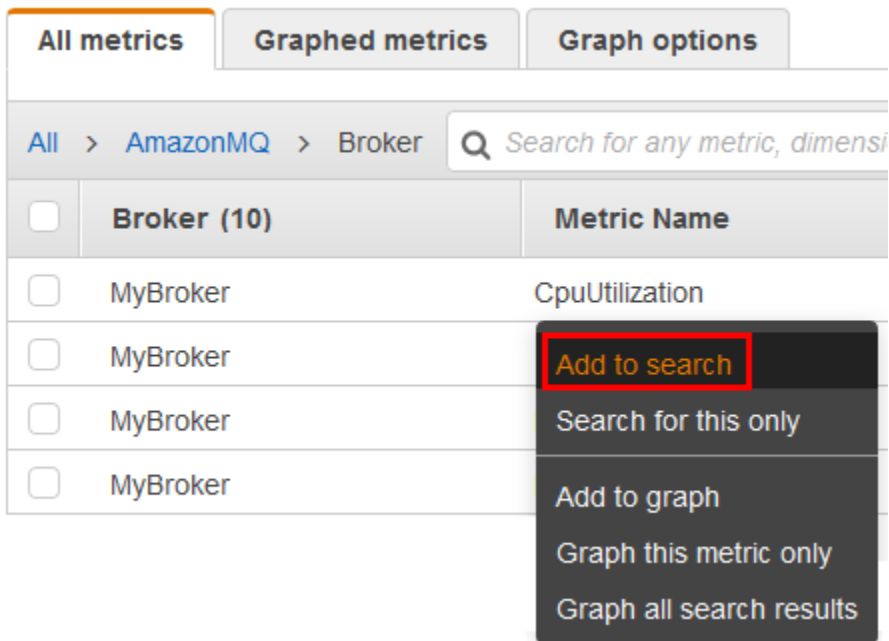
- Metrik Broker
- Metrik Antrian oleh Broker
- Metrik Topik oleh Broker

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.



## AWS Command Line Interface

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.

## CloudWatchAPI Amazon

Untuk mengakses metrik Amazon MQ menggunakan CloudWatch API, gunakan tindakan tersebut. [GetMetricStatistics](#)

Untuk informasi selengkapnya, lihat [Mendapatkan Statistik untuk Metrik](#) di Panduan CloudWatch Pengguna Amazon.

## Memantau broker Amazon MQ menggunakan Amazon CloudWatch

Amazon MQ dan Amazon CloudWatch terintegrasi sehingga Anda dapat menggunakan CloudWatch untuk melihat dan menganalisis metrik untuk broker ActiveMQ Anda dan tujuan broker (antrian dan topik). Anda dapat melihat dan menganalisis metrik Amazon MQ Anda dari CloudWatch konsol, AWS

CLI, atau CLI. CloudWatch CloudWatchmetrik untuk Amazon MQ secara otomatis disurvei dari broker dan kemudian didorong ke CloudWatch setiap menit.

Untuk informasi, lihat [Mengakses CloudWatch metrik untuk Amazon MQ](#).

#### Note

Statistik berikut ini berlaku untuk semua metrik:

- Average
- Minimum
- Maximum
- Sum

Namespace AWS/AmazonMQ mencakup metrik berikut.

Topik

- [Pencatatan dan pemantauan Amazon MQ untuk broker ActiveMQ](#)
- [Pencatatan dan pemantauan Amazon MQ untuk broker RabbitMQ](#)

## Pencatatan dan pemantauan Amazon MQ untuk broker ActiveMQ

### Metrik Amazon MQ untuk ActiveMQ

Metrik	Unit	Deskripsi
AmqpMaximumConnections	Count	Jumlah maksimum klien yang dapat Anda hubungkan ke broker Anda menggunakan AMQP. Untuk informasi lebih lanjut tentang kuota koneksi, lihat <a href="#">Quotas in Amazon MQ</a> .
BurstBalance	Persen	Persentase kredit burst yang tersisa pada volume Amazon EBS digunakan



Metrik	Unit	Deskripsi
		untuk menahan data pesan bagi broker yang dioptimalkan 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: <a href="#">Kredit I/O dan Performa Burst</a> .

Metrik	Unit	Deskripsi
CpuCreditBalance	Kredit (vCPU-menit)	<div data-bbox="1068 222 1508 682" style="border: 1px solid #f08080; border-radius: 10px; padding: 10px; margin-bottom: 10px;"> <p><b>⚠ Important</b></p> <p>Metrik ini hanya tersedia untuk tipe instans broker <code>mq.t2.micro</code>. Metrik kredit CPU hanya tersedia pada interval lima menit.</p> </div> <p>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.</p> <p>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.</p>
CpuUtilization	Persen	<p>Persentase unit komputasi Amazon EC2 yang dialokasikan dan saat ini digunakan oleh broker.</p>


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	Count	Jumlah maksimum klien yang dapat Anda hubungkan ke broker Anda menggunakan MQTT. Untuk informasi lebih lanjut tentang kuota koneksi, lihat <a href="#">Quotas in Amazon MQ</a> .

Metrik	Unit	Deskripsi
NetworkConnectorConnectionCount	Count	Jumlah node yang terhubung ke broker dalam <a href="#">jaringan broker</a> menggunakan Network Connector.
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	Count	Jumlah maksimum klien yang dapat Anda hubungkan ke broker Anda menggunakan OpenWire. Untuk informasi lebih lanjut tentang kuota koneksi, lihat <a href="#">Quotas in Amazon MQ</a> .
StompMaximumConnections	Count	Jumlah maksimum klien yang dapat Anda hubungkan ke broker Anda menggunakan STOMP. Untuk informasi lebih lanjut tentang kuota koneksi, lihat <a href="#">Quotas in Amazon MQ</a> .
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	Count	Jumlah maksimum klien yang dapat Anda hubungkan ke broker Anda menggunakan WebSocket. Untuk informasi lebih lanjut tentang kuota koneksi, lihat <a href="#">Quotas in Amazon MQ</a> .

### Dimensi untuk metrik broker ActiveMQ

Dimensi	Deskripsi
Broker	Nama broker

Dimensi	Deskripsi
	<div data-bbox="857 243 980 281">  Note         </div> <p data-bbox="906 302 1445 478">Broker instans tunggal memiliki sufiks -1. Broker aktif/siaga untuk ketersediaan tinggi memiliki sufiks -1 dan -2 untuk pasangan redundan.</p>

## Metrik tujuan ActiveMQ (antrean dan topik)


### Important


Metrik berikut mencakup hitungan per menit untuk periode pemungutan suara. CloudWatch


- EnqueueCount
- ExpiredCount
- DequeueCount
- DispatchCount
- InFlightCount

Misalnya, dalam [CloudWatchperiode](#) 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)	<p>Latensi menyeluruh dari saat pesan tiba di broker hingga dikirimkan ke konsumen.</p> <div data-bbox="1068 401 1507 1333" style="border: 1px solid #add8e6; border-radius: 10px; padding: 10px;"> <p> <b>Note</b></p> <p>EnqueueTime tidak mengukur latensi menyeluruh dari ketika pesan dikirim oleh produsen hingga mencapai broker, atau latensi dari ketika pesan diterima oleh broker hingga diakui oleh broker. Sebaliknya, EnqueueTime adalah jumlah milidetik dari saat pesan diterima oleh broker hingga berhasil dikirimkan ke konsumen.</p> </div>
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.
		<div style="border: 1px solid #f08080; border-radius: 10px; padding: 10px; background-color: #fff0f0;"> <p> <b>Important</b> Metrik ini hanya berlaku untuk antrean.</p> </div>
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.  <div style="border: 1px solid #add8e6; border-radius: 10px; padding: 10px; margin-top: 10px;"> <p> <b>Note</b>            Broker instans tunggal memiliki sufiks -1. Broker aktif/siaga untuk ketersediaan tinggi memiliki sufiks -1 dan -2 untuk pasangan redundan.</p> </div>
Topic atau Queue	Nama topik atau antrean.
NetworkConnector	Nama konektor jaringan.

## Pencatatan dan pemantauan 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.
ConsumerCount	Hitungan	Jumlah total konsumen yang terhubung ke broker.

Metrik	Unit	Deskripsi
MessageCount	Hitungan	<p>Jumlah total pesan dalam antrean.</p> <div style="border: 1px solid #add8e6; border-radius: 10px; padding: 10px; margin-top: 10px;"> <p> <b>Note</b></p> <p>Jumlah yang dihasilkan adalah jumlah total pesan siap dan tidak diakui pada broker.</p> </div>
MessageReadyCount	Count	Jumlah total pesan yang siap dalam antrean.
MessageUnacknowledgedCount	Hitungan	Jumlah total pesan yang tidak diakui dalam antrean.
PublishRate	Hitungan	<p>Laju saat pesan diterbitkan untuk broker.</p> <p>Jumlah yang dihasilkan mewakili jumlah pesan per detik pada saat pengambilan sampel.</p>
ConfirmRate	Hitungan	<p>Laju saat server RabbitMQ mengonfirmasi pesan yang diterbitkan. Anda dapat membandingkan metrik ini dengan PublishRate untuk lebih kinerja broker Anda.</p> <p>Jumlah yang dihasilkan mewakili jumlah pesan per detik pada saat pengambilan sampel.</p>

Metrik	Unit	Deskripsi
AckRate	Hitungan	<p>Laju saat pesan diakui oleh konsumen.</p> <p>Jumlah yang dihasilkan mewakili jumlah pesan per detik pada saat pengambilan sampel.</p>
SystemCpuUtilization	Persen	<p>Persentase unit komputasi Amazon EC2 yang dialokasikan dan saat ini digunakan oleh broker. Untuk penerapan klaster, nilai ini mewakili agregat dari ketiga nilai metrik yang sesuai dengan node RabbitMQ.</p>
RabbitMQMemLimit	Byte	<p>Batas RAM untuk broker RabbitMQ. Untuk penerapan klaster, nilai ini mewakili agregat dari ketiga nilai metrik yang sesuai dengan node RabbitMQ.</p>
RabbitMQMemUsed	Byte	<p>Volume RAM yang digunakan oleh broker RabbitMQ. Untuk penerapan klaster, nilai ini mewakili agregat dari ketiga nilai metrik yang sesuai dengan node RabbitMQ.</p>

Metrik	Unit	Deskripsi
RabbitMQDiskFreeLimit	Byte	Batas disk untuk broker RabbitMQ. Untuk penerapan klaster, nilai ini mewakili agregat dari ketiga nilai metrik yang sesuai dengan node RabbitMQ. Metrik ini berbeda per ukuran instans. Untuk informasi selengkapnya tentang jenis instans Amazon MQ, lihat <a href="#">the section called “Tipe instans Amazon MQ for RabbitMQ”</a> .
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 klaster, nilai ini mewakili agregat dari ketiga nilai metrik yang sesuai dengan node RabbitMQ.
RabbitMQFdUsed	Count	Jumlah deskriptor file yang digunakan. Untuk penerapan klaster, nilai ini mewakili agregat dari ketiga nilai metrik yang sesuai dengan node RabbitMQ.

Metrik	Unit	Deskripsi
RabbitMQIOReadAverageTime	Count	Waktu rata-rata (dalam milidetik) untuk RabbitMQ untuk melakukan satu operasi baca. Nilai sebanding dengan ukuran pesan.
RabbitMQIOWriteAverageTime	Count	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. Untuk informasi selengkapnya tentang jenis instans Amazon MQ, lihat <a href="#">the section called “Tipe instans Amazon MQ for RabbitMQ”</a> .
RabbitMQDiskFree	Byte	Volume total ruang disk kosong yang tersedia di simpul RabbitMQ. Ketika penggunaan disk melampaui batas, kluster akan memblokir semua koneksi produsen.
RabbitMQFdUsed	Hitungan	Jumlah deskriptor file yang digunakan.

## Dimensi untuk metrik simpul RabbitMQ

Dimensi	Deskripsi
Node	<p>Nama simpul.</p> <div style="border: 1px solid #ccc; border-radius: 10px; padding: 10px; margin-top: 10px;"> <p> <b>Note</b></p> <p>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</p> </div>

Dimensi	Deskripsi
	nama simpel dengan prefiks rabbit dan nama host ip-10-0-0-230.us-west-2.compute.internal .
Broker	Nama broker.

## Metrik antrean RabbitMQ

Metrik	Unit	Deskripsi
ConsumerCount	Hitungan	Jumlah konsumen berlangganan ke antrean.
MessageReadyCount	Count	Jumlah pesan yang saat ini tersedia untuk dikirimkan.
MessageUnacknowledgedCount	Hitungan	Jumlah pesan yang pengakuannya ditunggu oleh server.
MessageCount	Count	Jumlah total MessageReadyCount dan MessageUnacknowledgedCount (juga dikenal sebagai kedalaman antrean).

## Dimensi untuk metrik antrean RabbitMQ

### Note

Amazon MQ for RabbitMQ tidak akan mempublikasikan metrik untuk host virtual dan antrean dengan nama yang berisi spasi kosong, tab, atau karakter non-ASCII lainnya.

Untuk informasi selengkapnya tentang nama dimensi, lihat [Dimensi](#) dalam Referensi CloudWatch API Amazon.

Dimensi	Deskripsi
Queue	Nama antrian.
VirtualHost	Nama host virtual.
Broker	Nama broker.

## Pencatatan panggilan API Amazon MQ menggunakan AWS CloudTrail

Amazon MQ terintegrasi dengan AWS CloudTrail, layanan yang menyediakan catatan panggilan Amazon MQ yang dibuat pengguna, peran, atau AWS layanan. CloudTrail menangkap panggilan API yang terkait dengan broker Amazon MQ sebagai kejadian, termasuk panggilan dari konsol Amazon MQ dan panggilan kode ke API Amazon MQ. 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 Konsol Web ActiveMQ. Untuk mencatat informasi terkait operasi ActiveMQ, Anda dapat [mengonfigurasi Amazon MQ untuk memublikasikan log umum dan audit ke Amazon CloudWatch Logs](#).

Dengan menggunakan informasi yang CloudTrail dikumpulkan, Anda dapat mengidentifikasi permintaan khusus ke API Amazon MQ, alamat IP peminta, identitas peminta, tanggal serta waktu permintaan, dan sebagainya. Jika mengonfigurasi jejak, Anda dapat mengaktifkan pengiriman tindakan CloudTrail berkelanjutan ke bucket Amazon S3. Jika Anda tidak mengonfigurasi jejak, Anda dapat melihat tindakan terbaru di riwayat kejadian 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. Jika terjadi aktivitas kejadian Amazon MQ yang didukung, aktivitas tersebut dicatat dalam CloudTrail kejadian dengan tindakan lainnya dalam riwayat kejadian lainnya AWS di riwayat kejadian lainnya di riwayat kejadian. Anda dapat melihat, mencari, dan mengunduh kejadian terbaru untuk akun AWS Anda. Untuk informasi selengkapnya, lihat [Melihat CloudTrail Acara dengan Riwayat](#) Peristiwa di Panduan AWS CloudTrail Pengguna.

Jejak memungkinkan CloudTrail untuk mengirim berkas log ke bucket Amazon S3. Anda dapat membuat jejak untuk menyimpan catatan kejadian berkelanjutan di akun AWS. Secara default, ketika Anda membuat jejak di konsol menggunakan AWS Management Console, jejak tersebut diterapkan ke semua Wilayah AWS. Jejak mencatat kejadian dari semua Wilayah AWS dan mengirimkan berkas log ke bucket Amazon S3 yang ditentukan. Anda dapat mengonfigurasi AWS layanan lainnya untuk menganalisis lebih lanjut dan menindaklanjuti data peristiwa yang dikumpulkan di CloudTrail log. Untuk informasi lebih lanjut, lihat topik berikut di Panduan Pengguna AWS CloudTrail:

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

Amazon MQ mendukung pencatatan parameter permintaan dan respons untuk API berikut sebagai kejadian dalam berkas 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 metode GET API berikut, parameter permintaan dicatat, tetapi respons dihapus:

- [DescribeBroker](#)
- [DescribeConfiguration](#)
- [DescribeConfigurationRevision](#)
- [DescribeUser](#)
- [ListBrokers](#)
- [ListConfigurationRevisions](#)
- [ListConfigurations](#)
- [ListUsers](#)

Untuk API berikut, data dan parameter permintaan password disembunyikan dengan 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 dibuat oleh layanan AWS lainnya?

Untuk informasi selengkapnya, lihat [ElemenCloudTrail userIdentity](#) di PanduanAWS CloudTrail Pengguna.

## Contoh Entri Berkas Log Amazon MQ

Jejak adalah konfigurasi yang memungkinkan pengiriman kejadian sebagai berkas log ke bucket Amazon S3 yang ditentukan. CloudTrail berkas log berisi satu atau beberapa 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 [CreateBrokerAPI](#).

### Note

Karena berkas CloudTrail log bukan jejak tumpukan terurut dari API publik, berkas tersebut 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-3i45j6k17819",
    "brokerArn": "arn:aws:mq:us-east-2:123456789012:broker:MyBroker:b-1234a5b6-78cd-901e-2fgh-3i45j6k17819"
  },
  "requestID": "a1b2c345-6d78-90e1-f2g3-4hi56jk71890",
  "eventID": "a12bcd3e-fg45-67h8-ij90-12k34d5116mn",
  "readOnly": false,
  "eventType": "AwsApiCall",
  "recipientAccountId": "111122223333"
}
```

# Mengonfigurasi Amazon MQ untuk memublikasikan log ke Amazon CloudWatch Logs

Amazon MQ terintegrasi dengan Amazon CloudWatch Logs, layanan yang memantau, menyimpan, dan mengakses berkas log Anda dari berbagai sumber. Misalnya, Anda dapat [mengonfigurasi CloudWatch alarm](#) untuk menerima pemberitahuan [reboot broker](#) atau memecahkan masalah kesalahan [konfigurasi broker ActiveMQ](#). Untuk informasi selengkapnya tentang CloudWatch Log, lihat [Panduan Pengguna Amazon CloudWatch Logs](#)

## Topik

- [Mengonfigurasi Amazon MQ untuk ActiveMQ](#)
- [Mengonfigurasi Amazon MQ untuk RabbitMQ](#)

## Mengonfigurasi Amazon MQ untuk ActiveMQ

Untuk mengizinkan Amazon MQ memublikasikan CloudWatch log ke Logs, [Anda harus menambahkan izin ke pengguna Amazon MQ](#) dan juga [mengonfigurasi kebijakan berbasis sumber daya untuk Amazon MQ](#) sebelum membuat atau memulai ulang broker.

Berikut adalah penjelasan langkah-langkah mengonfigurasi CloudWatch log untuk broker ActiveMQ Anda.

## Topik

- [Memahami struktur pencatatan di CloudWatch Logs](#)
- [Tambahkan CreateLogGroup izin ke pengguna Amazon MQ Anda](#)
- [Mengonfigurasi kebijakan berbasis sumber daya untuk Amazon MQ](#)
- [Pencegahan wakil bingung lintas layanan](#)
- [Memecahkan Masalah Konfigurasi CloudWatch Logs](#)

## Memahami struktur pencatatan di CloudWatch Logs

Anda dapat mengaktifkan umum dan pencatatan audit saat [mengonfigurasi pengaturan broker lanjutan](#) ketika membuat broker, atau saat Anda mengedit broker.

Pencatatan umum memungkinkan tingkat INFO pencatatan default (DEBUG pencatatan tidak didukung) dan memublikasikan `activemq.log` ke grup log di CloudWatch akun Anda. Grup log memiliki format yang serupa dengan hal berikut:

```
/aws/amazonmq/broker/b-1234a5b6-78cd-901e-2fgh-3i45j6k17819/general
```

[Pencatatan audit](#) memungkinkan pencatatan tindakan manajemen yang diambil menggunakan JMX atau menggunakan Konsol Web ActiveMQ dan memublikasikan `audit.log` ke grup log di CloudWatch akun Anda. Grup log memiliki format yang serupa dengan hal berikut:

```
/aws/amazonmq/broker/b-1234a5b6-78cd-901e-2fgh-3i45j6k17819/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-3i45j6k17819-1.log  
activemq-b-1234a5b6-78cd-901e-2fgh-3i45j6k17819-2.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, Anda harus memastikan bahwa pengguna yang membuat atau melakukan boot ulang broker memiliki `logs:CreateLogGroup` izin.

### 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.

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

```
{
  "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 selengkapnya tentang menyiapkan pengguna dan mengonfigurasi kebijakan IAM, lihat bagian [Gambaran Umum Manajemen](#) entitas dalam 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 memublikasikan CloudWatch log ke Logs.

Untuk mengizinkan Amazon MQ memublikasikan CloudWatch log ke grup log Anda, konfigurasi kebijakan berbasis sumber daya untuk memberikan Amazon MQ akses ke tindakan API CloudWatch

- [CreateLogStream](#)— Membuat pengaliran CloudWatch log untuk grup log yang ditentukan.
- [PutLogEvents](#)— Mengirimkan kejadian ke pengaliran CloudWatch log yang ditentukan.

Kebijakan berbasis sumber daya berikut ini memberikan izin untuk `logs:CreateLogStream` dan `logs:PutLogEvents` ke AWS.

```
{
  "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 menggunakan AWS CLI seperti yang ditampilkan 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 prefiks `/aws/amazonmq/`, Anda perlu mengonfigurasi kebijakan berbasis sumber daya hanya satu kali per akun AWS, per wilayah.

## Pencegahan wakil bingung lintas layanan

Masalah wakil yang membingungkan adalah masalah keamanan di mana entitas yang tidak memiliki izin untuk melakukan tindakan dapat memaksa entitas yang lebih istimewa untuk melakukan tindakan. Dalam AWS, peniruan lintas layanan dapat mengakibatkan masalah wakil bingung. Peniruan lintas layanan dapat terjadi ketika satu layanan (layanan panggilan) memanggil layanan lain (layanan yang disebut). Layanan panggilan dapat dimanipulasi untuk menggunakan izinnya untuk bertindak atas sumber daya pelanggan lain dengan cara yang seharusnya tidak memiliki izin untuk mengakses. Untuk mencegah hal ini, AWS sediakan alat yang membantu Anda melindungi data Anda untuk semua layanan dengan prinsipal layanan yang telah diberikan akses ke sumber daya di akun Anda.

Kami merekomendasikan penggunaan [aws:SourceArn](#) dan kunci konteks kondisi [aws:SourceAccount](#) 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 broker Amazon MQ tunggal.

```
{
  "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-east-2:123456789012:broker:MyBroker:b-1234a5b6-78cd-901e-2fgh-3i45j6k17819"
        }
      }
    }
  ]
}
```

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.

```
{
  "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.

## Memecahkan Masalah Konfigurasi CloudWatch Logs

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.

### Pengaliran Log Tidak Muncul di Grup CloudWatch Log

[Konfigurasi kebijakan berbasis sumber daya untuk Amazon MQ](#). Hal ini memungkinkan broker Anda memublikasikan log-nya.

## Mengonfigurasi Amazon MQ untuk RabbitMQ

Saat Anda mengaktifkan CloudWatch pencatatan untuk broker RabbitMQ, Amazon MQ menggunakan peran terkait layanan untuk memublikasikan 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 memublikasikan log ke CloudWatch.

Untuk informasi selengkapnya tentang peran terkait layanan, lihat [Menggunakan peran terkait layanan](#) dalam 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”](#).

# Kuota di Amazon MQ

Topik ini mencantumkan kuota dalam Amazon MQ. Banyak kuota berikut dapat diubah untuk akun AWS spesifik. Untuk meminta kenaikan batas, lihat [Kuota AWS Layanan](#) di bagian. 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 AmazonCloudWatch, 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	<ul style="list-style-type: none"><li>• Harus unik dalam akun AWS Anda.</li><li>• Harus terdiri dari 1-50 karakter.</li><li>• Hanya dapat berisi karakter yang ditentukan dalam <a href="#">Set Karakter ASCII yang Dapat Dicitak</a>.</li><li>• Hanya dapat berisi karakter alfanumerik, tanda hubung, titik, garis bawah, dan tilde (- . _ ~).</li></ul>
Jumlah broker, per wilayah	50

Kuota	Deskripsi
Koneksi tingkat kawat per protokol untuk broker yang lebih kecil	<div style="border: 1px solid #f08080; border-radius: 10px; padding: 10px; margin-bottom: 10px;">  <b>Important</b> Tidak berlaku untuk broker RabbitMQ.         </div> <p>300 mq.*.micro misalnya jenis broker.</p>
Koneksi tingkat kawat per protokol untuk broker yang lebih besar	<div style="border: 1px solid #f08080; border-radius: 10px; padding: 10px; margin-bottom: 10px;">  <b>Important</b> Tidak berlaku untuk broker RabbitMQ.         </div> <p>2.000 mq.*.*large misalnya jenis broker.</p>
Grup keamanan per broker	5
Tujuan ActiveMQ (antrian, dan topik) dipantau di CloudWatch	CloudWatchmemonitor hanya 1000 tujuan pertama.
Tujuan RabbitMQ (antrian) dipantau di CloudWatch	CloudWatchmemonitor hanya 500 tujuan pertama, dipesan oleh jumlah konsumen.
Tanda per broker	50

## Konfigurasi

Tabel berikut mencantumkan kuota yang terkait dengan konfigurasi Amazon MQ.

 **Important**  
Tidak berlaku untuk broker RabbitMQ.

Kuota	Deskripsi
Nama konfigurasi	•

Kuota	Deskripsi
	<p>Harus terdiri dari 1-150 karakter.</p> <ul style="list-style-type: none"> <li>• Hanya dapat berisi karakter yang ditentukan dalam <a href="#">Set Karakter ASCII yang Dapat Dicitak</a>.</li> <li>• Hanya dapat berisi karakter alfanumerik, tanda hubung, titik, garis bawah, dan tilde (- . _ ~).</li> </ul>
Revisi per konfigurasi	300

## Pengguna

Tabel berikut mencantumkan kuota yang terkait dengan pengguna broker ActiveMQ Amazon MQ.

### Important


Tidak berlaku untuk broker RabbitMQ.



Kuota	Deskripsi
Nama pengguna	<ul style="list-style-type: none"> <li>• Harus terdiri dari 1-100 karakter.</li> <li>• Hanya dapat berisi karakter yang ditentukan dalam <a href="#">Set Karakter ASCII yang Dapat Dicitak</a>.</li> <li>• Hanya dapat berisi karakter alfanumerik, tanda hubung, titik, garis bawah, dan tilde (- . _ ~).</li> <li>• Tidak boleh berisi tanda koma (,).</li> </ul>

Kuota	Deskripsi
Kata Sandi	<ul style="list-style-type: none"> <li>• Harus terdiri dari 12-250 karakter.</li> <li>• Hanya dapat berisi karakter yang ditentukan dalam <a href="#">Set Karakter ASCII yang Dapat Dicitak</a>.</li> <li>• Harus terdiri dari setidaknya 4 karakter unik.</li> <li>• Tidak boleh berisi tanda koma (,).</li> </ul>
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 <a href="#">Broker instance types</a> .
Kapasitas penyimpanan per broker yang lebih besar	200 GB untuk broker tipe instans mq.*.*large. Untuk informasi selengkapnya mengenai tipe instans Amazon MQ, lihat <a href="#">Broker instance types</a> .
Batasan penggunaan penjadwal tugas per broker <a href="#">didukung Amazon EBS</a>	<div style="border: 1px solid #f08080; border-radius: 10px; padding: 10px;"> <p> Important</p> <p>Tidak berlaku untuk broker RabbitMQ.</p> </div>

Kuota	Deskripsi
	50 GB. Untuk informasi selengkapnya tentang penggunaan penjadwal tugas, lihat <a href="#">JobSchedulerUsage</a> dalam Dokumentasi API Apache ActiveMQ.
Kapasitas penyimpanan sementara per broker yang lebih kecil.	<div data-bbox="829 464 1507 636" style="border: 1px solid #f08080; border-radius: 10px; padding: 10px; margin-bottom: 10px;"> <p> <b>Important</b> Tidak berlaku untuk broker RabbitMQ.</p> </div> <p>5 GB untuk broker tipe instans mq.*.micro .</p>
Kapasitas penyimpanan sementara per broker yang lebih besar.	<div data-bbox="829 812 1507 984" style="border: 1px solid #f08080; border-radius: 10px; padding: 10px; margin-bottom: 10px;"> <p> <b>Important</b> Tidak berlaku untuk broker RabbitMQ.</p> </div> <p>50 GB untuk broker tipe instans mq.*.*large .</p>

## Throttling API

Kuota throttling berikut diagregat per akun AWS, di semua API Amazon MQ untuk mempertahankan bandwidth layanan. Untuk informasi selengkapnya tentang API Amazon MQ, lihat [Referensi API REST Amazon MQ](#).

 **Important**

Kuota ini tidak berlaku untuk API olahpesan broker Amazon MQ for ActiveMQ atau Amazon MQ for RabbitMQ. 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.

## Daftar Isi

- [Pemecahan masalah: Umum](#)
  - [Saya tidak dapat terhubung ke konsol web broker atau titik akhir saya.](#)
  - [Broker saya sedang berjalan, dan saya dapat memverifikasi konektivitas menggunakan telnet, 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.](#)
- [Pemecahan masalah: Amazon MQ untuk ActiveMQ](#)
  - [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 statusnya RUNNING. 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.](#)
- [Pemecahan masalah: Amazon MQ untuk RabbitMQ](#)
  - [Saya tidak dapat melihat metrik untuk antrian atau host virtual saya di CloudWatch](#)
  - [Bagaimana cara mengaktifkan plugin di Amazon MQ untuk RabbitMQ?](#)
  - [Saya tidak dapat mengubah konfigurasi VPC Amazon untuk broker.](#)
- [Pemecahan masalah: Kode yang diperlukan tindakan Amazon MQ](#)
  - [Amazon MQ untuk RabbitMQ: Alarm memori tinggi](#)
    - [Mendiagnosis alarm memori tinggi menggunakan konsol web RabbitMQ](#)
    - [Mendiagnosis alarm memori tinggi menggunakan metrik Amazon MQ](#)
    - [Mengatasi alarm memori tinggi](#)
    - [Mengurangi jumlah koneksi dan saluran](#)
    - [Mengatasi sinkronisasi antrian yang dijeda dalam penerapan kluster](#)

- [Mengatasi loop restart di broker satu instans](#)
- [Mencegah alarm memori tinggi](#)
- [Amazon MQ untuk RabbitMQ: Kunci Tidak Valid AWS Key Management Service](#)
  - [Mendiagnosis dan menangani INVALID\\_KMS\\_KEY](#)
- [Amazon MQ untuk ActiveMQ: Alarm Antarmuka Jaringan Elastis yang Dihapus](#)
- [Amazon MQ untuk ActiveMQ: Broker Kehabisan Alarm Memori](#)
- [Amazon MQ untuk RabbitMQ: Alarm batas disk](#)
  - [Mendiagnosis dan menangani alarm batas disk](#)

## Pemecahan masalah: 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 menggunakan telnet, 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 ACL

jaringan khusus, 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 Amazon MQ untuk ActiveMQ dan Amazon MQ untuk RabbitMQ menggunakan port yang berbeda untuk koneksi.


#### Amazon MQ for ActiveMQ

- Konsol web — Pelabuhan 8162
- OpenWire — Pelabuhan 61617
- AMQP - Pelabuhan 5671
- STOMP - Pelabuhan 61614
- MQTT - Pelabuhan 8883
- WSS - Pelabuhan 61619

#### Amazon MQ for RabbitMQ

- 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.

#### Amazon MQ for ActiveMQ

Untuk menguji Amazon MQ Anda untuk konektivitas jaringan broker ActiveMQ

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. Titik akhir aktif/siaga 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 penerapan aktif/siaga, 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-3i45j6k17819-1.mq.us-  
west-2.amazonaws.com.  
Escape character is '^]'.
```

4. Lakukan salah satu dari berikut ini.

- Jika `telnet` perintah berhasil, periksa [EstablishedConnectionsCount](#) metrik dan konfirmasi 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 konfirmasi bahwa statusnya adalah `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 konfirmasi 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-3i45j6k17819-1.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>
```

...

## Amazon MQ for RabbitMQ

Untuk menguji Amazon MQ Anda untuk konektivitas jaringan broker RabbitMQ

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-3i45j6k17819-1.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-3i45j6k17819-1.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-3i45j6k17819-1.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-3i45j6k17819-1.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 konfirmasi 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 konfirmasi 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 konfirmasi paket koneksiACCEPTED, 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 Amazon MQ untuk broker RabbitMQ dengan aksesibilitas publik.

## 5. Jalankan curl perintah berikut untuk memeriksa konektivitas ke konsol web admin RabbitMQ.

```
$ curl https://b-1234a5b6-78cd-901e-2fgh-3i45j6k17819-1.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, lihat [the section called “Mempertahankan broker”](#).
- Jenis instans broker Anda mungkin tidak sesuai dengan beban kerja aplikasi Anda. Misalnya, menjalankan beban kerja produksi pada `a mq.t2.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.
  - Amazon MQ untuk ActiveMQ - Periksa **CpuUtilization** persentase unit komputasi Amazon EC2 yang dialokasikan yang saat ini digunakan broker. Periksa **HeapUsage** persentase batas memori ActiveMQ JVM yang digunakan broker saat ini.
  - Amazon MQ untuk RabbitMQ — 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 selengkapnya tentang jenis instans broker dan cara memilih jenis instans yang tepat untuk beban kerja Anda, lihat [Broker instance types](#).

## Pemecahan masalah: Amazon MQ untuk ActiveMQ

Gunakan informasi di bagian ini untuk membantu Anda mendiagnosis dan menyelesaikan masalah umum yang mungkin Anda temui saat bekerja dengan Amazon MQ untuk broker ActiveMQ.

### 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 statusnya RUNNING. 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, konfigurasi 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 Amazon MQ untuk ActiveMQ untuk memublikasikan log CloudWatch 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 Amazon MQ Anda untuk broker ActiveMQ di Log. CloudWatch Jika batas telah tercapai, Anda akan melihat `Reached Maximum Connections` di log broker. Untuk informasi lebih lanjut tentang CloudWatch Log untuk Amazon MQ untuk broker ActiveMQ, lihat [the section called “Memahami struktur pencatatan di CloudWatch Logs”](#)

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, lihat [Broker 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 selengkapnya tentang integrasi LDAP, lihat [the section called "Cara kerja integrasi LDAP"](#).

## Pemecahan masalah: Amazon MQ untuk RabbitMQ

Gunakan informasi di bagian ini untuk membantu Anda mendiagnosis dan menyelesaikan masalah umum yang mungkin Anda temui saat bekerja dengan Amazon MQ untuk broker RabbitMQ.

### Daftar Isi

- [Saya tidak dapat melihat metrik untuk antrian atau host virtual saya di CloudWatch](#)
- [Bagaimana cara mengaktifkan plugin di Amazon MQ untuk RabbitMQ?](#)
- [Saya tidak dapat mengubah konfigurasi VPC Amazon untuk broker.](#)

## 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 Amazon MQ untuk RabbitMQ?

Amazon MQ untuk RabbitMQ saat ini hanya mendukung manajemen RabbitMQ, sekop, federasi, plugin pertukaran hash konsisten, yang diaktifkan secara default. Untuk informasi selengkapnya tentang penggunaan plugin yang didukung, lihat [the 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.

## Pemecahan masalah: Kode yang diperlukan tindakan Amazon MQ

Amazon MQ mengembalikan pengecualian untuk operasi API tertentu, seperti [RebootBroker](#), jika broker Anda dalam keadaan tidak sehat dan memerlukan serangkaian tindakan untuk kembali ke keadaan sehat. Pengecualian termasuk kode tindakan khusus yang diperlukan yang membantu Anda mengidentifikasi akar penyebab, dan mengatasi masalah dan memulihkan broker Anda.

Gunakan daftar topik berikut untuk mengidentifikasi kode tindakan yang diperlukan yang telah Anda terima, dan pelajari lebih lanjut tentang langkah-langkah yang kami sarankan untuk menyelesaikan masalah Anda.

Kode tindakan yang diperlukan

- [Amazon MQ untuk RabbitMQ: Alarm memori tinggi](#)
- [Amazon MQ untuk RabbitMQ: Kunci Tidak Valid AWS Key Management Service](#)
- [Amazon MQ untuk ActiveMQ: Alarm Antarmuka Jaringan Elastis yang Dihapus](#)

- [Amazon MQ untuk ActiveMQ: Broker Kehabisan Alarm Memori](#)
- [Amazon MQ untuk RabbitMQ: Alarm batas disk](#)

## Amazon MQ untuk RabbitMQ: Alarm memori tinggi

RabbitMQ akan meningkatkan alarm memori tinggi ketika penggunaan memori broker, diidentifikasi dengan CloudWatch metrik `RabbitMQMemUsed`, melebihi batas memori, diidentifikasi oleh `RabbitMQMemLimit`. `RabbitMQMemLimit` diatur oleh Amazon MQ dan telah disetel secara khusus mengingat memori yang tersedia untuk setiap jenis instance host.

Amazon MQ untuk broker RabbitMQ yang telah meningkatkan alarm memori tinggi akan memblokir semua klien yang menerbitkan pesan. Karena penggunaan memori yang tinggi, broker Anda mungkin juga mengalami masalah lain yang mempersulit diagnosis dan resolusi alarm.

Amazon MQ untuk broker RabbitMQ yang telah meningkatkan alarm memori tinggi akan memblokir semua klien yang menerbitkan pesan. Karena penggunaan memori yang tinggi, broker Anda mungkin juga mengalami masalah lain yang mempersulit diagnosis dan resolusi alarm.

Pialang instans tunggal yang tidak dapat menyelesaikan start-up karena penggunaan memori yang tinggi mungkin memasuki loop restart, di mana interaksi dengan broker terbatas. Dalam penerapan cluster, antrian mungkin mengalami sinkronisasi pesan yang dijeda antara replika pada node yang berbeda. Sinkronisasi antrian yang dijeda mencegah konsumsi pesan dari antrian dan harus ditangani secara terpisah saat menyelesaikan alarm memori.

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.

Gunakan informasi di bagian ini untuk membantu Anda mendiagnosis dan menyelesaikan alarm memori tinggi RabbitMQ yang dimunculkan oleh broker Anda.

### Note

Mungkin diperlukan waktu hingga beberapa jam agar status `RABBITMQ_MEMORY_ALARM` dihapus setelah Anda mengambil tindakan yang diperlukan.

### 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.

## Topik

- [Mendiagnosis alarm memori tinggi menggunakan konsol web RabbitMQ](#)
- [Mendiagnosis alarm memori tinggi menggunakan metrik Amazon MQ](#)
- [Mengatasi alarm memori tinggi](#)
- [Mengurangi jumlah koneksi dan saluran](#)
- [Mengatasi sinkronisasi antrian yang dijeda dalam penerapan kluster](#)
- [Mengatasi loop restart di broker satu instans](#)
- [Mencegah alarm memori tinggi](#)

## 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 AWS Management Console 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 yang tinggi	
MessageCount	Pesan disimpan dalam memori sampai dikonsumsi	



Metrik Amazon MQ CloudWatch	Alasan penggunaan memori yang tinggi	
	<p>i atau dibuang. Jumlah pesan yang tinggi mungkin menunjukkan pemanfaatan sumber daya yang berlebihan dan dapat menyebabkan alarm memori yang tinggi.</p>	
QueueCount	<p>Antrian disimpan dalam memori, dan sejumlah besar antrian dapat menyebabkan alarm memori yang tinggi.</p>	
ConnectionCount	<p>Koneksi klien menggunakan memori, dan terlalu banyak koneksi simultan dapat menyebabkan alarm memori tinggi.</p>	
ChannelCount	<p>Mirip dengan koneksi, saluran yang dibuat menggunakan setiap koneksi juga disimpan dalam memori node, dan sejumlah besar saluran dapat menyebabkan alarm memori yang tinggi.</p>	

Metrik Amazon MQ CloudWatch	Alasan penggunaan memori yang 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 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.	

## Mengatasi alarm memori tinggi

Untuk setiap kontributor yang Anda identifikasi, kami merekomendasikan serangkaian tindakan berikut untuk mengurangi dan menyelesaikan alarm memori tinggi broker.

Alasan penggunaan memori yang tinggi	Rekomendasi Amazon MQ	
Jumlah pesan dalam antrian terlalu tinggi.	Lakukan langkah-langkah berikut:	

Alasan penggunaan memori yang tinggi	Rekomendasi Amazon MQ	
	<ul style="list-style-type: none"><li>• Konsumsi pesan yang dipublikasikan ke antrian.</li><li>• Bersihkan pesan dari antrian.</li><li>• Hapus antrian dari broker Anda.</li></ul>	
Jumlah antrian yang dikonfigurasi pada broker terlalu tinggi.	Kurangi jumlah antrian.	
Jumlah koneksi yang dibuat pada broker terlalu tinggi.	Kurangi jumlah koneksi. Untuk informasi selengkapnya, lihat <a href="#">the section called “Mengurangi jumlah koneksi dan saluran”</a> .	
Jumlah saluran yang didirikan pada broker terlalu tinggi.	Kurangi jumlah saluran. Untuk informasi lebih lanjut lihat, <a href="#">the section called “Mengurangi jumlah koneksi dan saluran”</a> .	
Jumlah konsumen yang terhubung ke broker terlalu tinggi.	Kurangi jumlah konsumen yang terhubung ke broker.	
Tingkat penerbitan pesan terlalu tinggi.	Kurangi tingkat di mana penerbit mengirim pesan ke broker.	
Tingkat upaya koneksi klien terlalu tinggi.	Kurangi frekuensi di mana klien mencoba untuk terhubung ke broker untuk mempublikasikan atau mengkonsumsi pesan, atau mengkonfigurasi broker.	

## Mengurangi jumlah koneksi dan saluran

Koneksi ke Amazon MQ Anda untuk broker RabbitMQ 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 AWS Management Console 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. Amazon MQ untuk RabbitMQ 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.

## Mengatasi sinkronisasi antrian yang dijeda dalam penerapan klaster

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 antrian yang dijeda”](#)

## Mengatasi loop restart di broker satu instans

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 tindakan yang direkomendasikan Amazon MQ yang sebelumnya dijelaskan di bagian ini 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.

## Mencegah alarm memori tinggi

Untuk setiap faktor yang berkontribusi yang Anda identifikasi, kami merekomendasikan serangkaian tindakan berikut untuk mencegah dan mengurangi terjadinya alarm memori tinggi RabbitMQ.

Alasan penggunaan memori tinggi	Rekomendasi Amazon MQ
Jumlah pesan dalam antrian terlalu tinggi.	Lakukan hal berikut: <ul style="list-style-type: none"> <li>Aktifkan <a href="#">antrian malas</a>.</li> <li>Atur, atau kurangi <a href="#">batas kedalaman antrian</a>.</li> </ul>
Jumlah antrian yang dikonfigurasi pada broker terlalu tinggi.	Tetapkan, atau kurangi <a href="#">batas hitungan antrian</a> .
Jumlah koneksi yang dibuat pada broker terlalu tinggi.	Atur, atau kurangi <a href="#">batas jumlah koneksi</a> .

Alasan penggunaan memori tinggi	Rekomendasi Amazon MQ
Jumlah saluran yang didirikan pada broker terlalu tinggi.	Tetapkan jumlah maksimum saluran per koneksi pada aplikasi klien.
Jumlah konsumen yang terhubung ke broker terlalu tinggi.	Tetapkan <a href="#">batas pra-pengambilan</a> konsumen yang kecil.
Tingkat upaya koneksi klien terlalu tinggi.	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.

Untuk daftar lengkap jenis instans broker, lihat [the section called “Tipe instans Amazon MQ for RabbitMQ”](#).

## Amazon MQ untuk RabbitMQ: Kunci Tidak Valid AWS Key Management Service

Amazon MQ untuk RabbitMQ 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

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` 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.

## Amazon MQ untuk ActiveMQ: Alarm Antarmuka Jaringan Elastis yang Dihapus

Amazon MQ untuk ActiveMQ 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.

## Amazon MQ untuk ActiveMQ: Broker Kehabisan Alarm Memori

Amazon MQ untuk ActiveMQ 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 yang tinggi
<code>TotalMessageCount</code>	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.
<code>HeapUsage</code>	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.
<code>ConnectionCount</code>	Koneksi klien menggunakan memori, dan terlalu banyak



Metrik Amazon MQ CloudWatch	Alasan penggunaan memori yang tinggi	
	koneksi simultan dapat 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 penyimpanan ke memori sebelum dikirim ke konsumen. Sejumlah besar koneksi konsumen dapat menyebabkan penggunaan 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 [Amazon MQ untuk praktik terbaik ActiveMQ](#).

## Amazon MQ untuk RabbitMQ: 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 AmazonRabbitMQDiskFree, 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.

Amazon MQ untuk broker RabbitMQ yang telah menaikkan alarm batas disk akan menjadi tidak tersedia untuk pesan baru yang diterbitkan. 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:

- Konsumsi 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.

# Sumber daya terkait

## Sumber daya Amazon MQ

Tabel berikut mencantumkan sumber daya yang bermanfaat untuk bekerja dengan Amazon MQ.

Sumber Daya	Deskripsi
<a href="#">Referensi REST API Amazon MQ</a>	Deskripsi sumber daya REST, permintaan contoh, metode HTTP, skema, parameter, dan kesalahan yang dikembalikan layanan.
<a href="#">Amazon MQ diAWS CLIReferensi Perintah</a>	Deskripsi perintah AWS CLI yang dapat Anda gunakan untuk bekerja dengan broker pesan.
<a href="#">Amazon MQ diAWS CloudFormationPanduan Pengguna</a>	<p>Sumber daya <a href="#">AWS::Amazon MQ::Broker</a> memungkinkan Anda membuat broker Amazon MQ, menambahkan perubahan konfigurasi atau memodifikasi pengguna untuk broker tertentu, mengembalikan informasi tentang broker tertentu, dan menghapus broker tertentu.</p> <p>Sumber daya <a href="#">AWS::Amazon MQ::Configuration</a> memungkinkan Anda membuat konfigurasi Amazon MQ, menambahkan perubahan konfigurasi atau memodifikasi pengguna, dan mengembalikan informasi tentang konfigurasi tertentu.</p>
<a href="#">Wilayah dan Titik Akhir</a>	Informasi tentang wilayah dan titik akhir Amazon MQ
<a href="#">Halaman Produk</a>	Halaman web utama untuk informasi tentang Amazon MQ.

Sumber Daya	Deskripsi
<a href="#">Forum Diskusi</a>	Forum berbasis komunitas untuk developer guna membahas pertanyaan teknis terkait Amazon MQ.
<a href="#">AWSInformasi Premium Support</a>	Halaman web utama untuk informasi tentang Premium Support AWS, saluran dukungan respons cepat langsung untuk membantu Anda membangun dan menjalankan aplikasi di layanan infrastruktur AWS

## Sumber daya Amazon MQ for ActiveMQ

Tabel berikut mencantumkan sumber daya yang bermanfaat untuk bekerja dengan Apache ActiveMQ.

Sumber Daya	Deskripsi
<a href="#">Panduan Memulai dengan Apache ActiveMQ</a>	Dokumentasi resmi dari Apache ActiveMQ.
<a href="#">ActiveMQ dalam Tindakan</a>	Panduan untuk Apache ActiveMQ yang mencakup anatomi pesan, konektor, persistensi pesan, autentikasi, dan otorisasi JMS.
<a href="#">Klien Lintas Bahasa</a>	Daftar bahasa pemrograman dan pustaka Apache ActiveMQ yang sesuai. Lihat juga <a href="#">Klien ActiveMQ</a> dan <a href="#">Klien QpidJMS</a> .

## Sumber daya Amazon MQ for RabbitMQ

Tabel berikut mencantumkan sumber daya yang bermanfaat untuk bekerja dengan RabbitMQ.

Sumber Daya	Deskripsi
<a href="#">Panduan Memulai dengan RabbitMQ</a>	Dokumentasi resmi RabbitMQ.

Sumber Daya	Deskripsi
<a href="#">Pustaka Klien RabbitMQ dan Alat Developer</a>	Panduan untuk pustaka klien yang didukung secara resmi dan alat developer untuk bekerja dengan RabbitMQ menggunakan berbagai bahasa pemrograman dan platform.
<a href="#">Praktik Terbaik RabbitMQ</a>	Panduan CloudAMQP tentang praktik terbaik dan rekomendasi untuk bekerja dengan RabbitMQ.

# Catatan rilis Amazon MQ

Tabel berikut mencantumkan rilis fitur dan peningkatan Amazon MQ. Untuk perubahan pada Panduan Developer Amazon MQ, lihat [Riwayat Dokumen Amazon MQ](#).

Tanggal	Pembaruan Dokumentasi
Maret 4, 2023	<p>Amazon MQ untuk RabbitMQ sekarang mendukung RabbitMQ 3.11.28.</p> <p>Untuk informasi selengkapnya tentang perbaikan dan fitur dalam rilis ini, lihat berikut ini:</p> <ul style="list-style-type: none"><li>• <a href="#">Catatan rilis RabbitMQ 3.11.28 pada repositori server RabbitMQ</a> GitHub</li><li>• <a href="#">RabbitMQ changelog</a></li></ul> <p>Untuk informasi selengkapnya tentang Amazon MQ yang didukung untuk versi RabbitMQ dan peningkatan broker, lihat <a href="#">Mengelola Amazon MQ untuk versi mesin RabbitMQ</a></p>
Januari 19, 2024	<p>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”.</p>
15 Desember 2023	<p>Amazon MQ sekarang tersedia di Wilayah Israel (Tel Aviv). Untuk informasi tentang wilayah yang tersedia, lihat <a href="#">AWS Wilayah dan Titik Akhir</a> dalam panduan Referensi AWS Umum.</p>
Desember 11, 2023	<p>Amazon MQ untuk RabbitMQ sekarang mendukung RabbitMQ 3.10.25.</p> <p>Untuk informasi selengkapnya tentang perbaikan dan fitur dalam rilis ini, lihat berikut ini:</p> <ul style="list-style-type: none"><li>• <a href="#">Catatan rilis RabbitMQ 3.10.25 pada repositori server RabbitMQ</a> GitHub</li><li>• <a href="#">RabbitMQ changelog</a></li></ul>

Tanggal	Pembaruan Dokumentasi
26 Oktober 2023	<p>Untuk informasi selengkapnya tentang Amazon MQ yang didukung untuk versi RabbitMQ dan peningkatan broker, lihat. <a href="#">Mengelola Amazon MQ untuk versi mesin RabbitMQ</a></p> <p>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.</p> <p>Untuk informasi lebih lanjut tentang memperbarui broker ActiveMQ Anda, lihat. <a href="#">Mengelola versi mesin Amazon MQ for ActiveMQ</a></p>
27 September 2023	<p>Amazon MQ untuk RabbitMQ sekarang mendukung RabbitMQ 3.11.20.</p> <p>Untuk informasi selengkapnya tentang perbaikan dan fitur dalam rilis ini, lihat berikut ini:</p> <ul style="list-style-type: none"><li>• <a href="#">Catatan rilis RabbitMQ 3.11.20 pada repositori server RabbitMQ</a> GitHub</li><li>• <a href="#">RabbitMQ changelog</a></li></ul> <p>Untuk informasi selengkapnya tentang Amazon MQ yang didukung untuk versi RabbitMQ dan peningkatan broker, lihat. <a href="#">Mengelola Amazon MQ untuk versi mesin RabbitMQ</a></p>
Juli 27, 2023	<p>Amazon MQ untuk RabbitMQ sekarang mendukung RabbitMQ 3.11.16</p> <p>Untuk informasi selengkapnya tentang perbaikan dan fitur dalam rilis ini, lihat berikut ini:</p> <ul style="list-style-type: none"><li>• <a href="#">Catatan rilis RabbitMQ 3.11.16 pada repositori server RabbitMQ</a> GitHub</li><li>• <a href="#">RabbitMQ changelog</a></li></ul> <p>Untuk informasi selengkapnya tentang Amazon MQ yang didukung untuk versi RabbitMQ dan peningkatan broker, lihat. <a href="#">Mengelola Amazon MQ untuk versi mesin RabbitMQ</a></p>



Tanggal	Pembaruan Dokumentasi
Juli 27, 2023	<p>Amazon MQ untuk RabbitMQ sekarang mendukung pembuatan dan penerapan konfigurasi ke broker RabbitMQ Anda.</p> <p>Untuk informasi lebih lanjut tentang menambahkan konfigurasi ke broker Anda, lihat <a href="#">RabbitMQ Broker Configurations</a></p> <p>Untuk informasi selengkapnya tentang fitur ini, lihat:</p> <ul style="list-style-type: none"><li>• <a href="#">Kebijakan operator</a></li><li>• <a href="#">Perubahan kebijakan operator</a></li></ul>
23 Juni 2023	<p>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.</p> <p>Untuk informasi selengkapnya, lihat hal berikut:</p> <ul style="list-style-type: none"><li>• Untuk memulai CRDR, lihat <a href="#">Replikasi data Lintas Wilayah untuk Amazon MQ untuk ActiveMQ</a> di Panduan Pengembang.</li><li>• <a href="#">ActiveMQ 5.17.3 Halaman Rilis</a></li><li>• <a href="#">Mengelola versi mesin Amazon MQ for ActiveMQ</a></li><li>• <a href="#">Meningkatkan versi mesin broker Amazon MQ</a></li><li>• <a href="#">Bekerja dengan file konfigurasi Spring XML</a></li></ul>
Juni 21, 2023	<p>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.</p> <p>Untuk memulai CRDR, lihat <a href="#">Replikasi data Lintas Wilayah untuk Amazon MQ untuk ActiveMQ</a> di Panduan Pengembang.</p>

Tanggal	Pembaruan Dokumentasi
18 Mei 2023	<p>Amazon MQ sekarang tersedia di wilayah berikut:</p> <ul style="list-style-type: none"><li>• Asia Pasifik (Melbourne)</li><li>• Asia Pasifik (Hyderabad)</li><li>• Eropa (Spanyol)</li><li>• Eropa (Zurich)</li></ul> <p>Untuk informasi tentang wilayah yang tersedia, lihat <a href="#">AWS Wilayah dan Titik Akhir</a> dalam panduan Referensi AWS Umum.</p>
April 14, 2023	<p>Amazon MQ untuk RabbitMQ sekarang mendukung RabbitMQ versi 3.9.27.</p> <p>Untuk informasi selengkapnya tentang perbaikan dan fitur dalam rilis ini, lihat berikut ini:</p> <ul style="list-style-type: none"><li>• <a href="#">Catatan rilis RabbitMQ 3.9.27 pada repositori server RabbitMQ</a> GitHub</li><li>• <a href="#">RabbitMQ changelog</a></li></ul> <p>Untuk informasi selengkapnya tentang Amazon MQ yang didukung untuk versi RabbitMQ dan peningkatan broker, lihat. <a href="#">Mengelola Amazon MQ untuk versi mesin RabbitMQ</a></p>
April 14, 2023	<p>Amazon MQ untuk RabbitMQ sekarang mendukung RabbitMQ versi 3.10.20.</p> <p>Untuk informasi selengkapnya tentang perbaikan dan fitur dalam rilis ini, lihat berikut ini:</p> <ul style="list-style-type: none"><li>• <a href="#">Catatan rilis RabbitMQ 3.10.20 pada repositori server RabbitMQ</a> GitHub</li><li>• <a href="#">RabbitMQ changelog</a></li></ul> <p>Untuk informasi selengkapnya tentang Amazon MQ yang didukung untuk versi RabbitMQ dan peningkatan broker, lihat. <a href="#">Mengelola Amazon MQ untuk versi mesin RabbitMQ</a></p>


Tanggal	Pembaruan Dokumentasi
31 Maret 2023	<p>Amazon MQ untuk RabbitMQ telah menonaktifkan mesin RabbitMQ versi 3.10.17</p> <p><a href="#">Amazon MQ untuk tim RabbitMQ, dan pengelola open source RabbitMQ, telah mengidentifikasi masalah dengan konsol manajemen RabbitMQ pada versi 3.10.17.</a> Amazon MQ telah mencabut versi ini. Untuk mengurangi dampak dari masalah ini, buat broker baru dengan versi 3.10.20 sementara kami bekerja untuk mendukung versi patch baru RabbitMQ. Sebaiknya aktifkan opsi <a href="#">pemutakhiran versi minor otomatis</a> untuk secara otomatis mendapatkan perbaikan bug terbaru, pembaruan keamanan, dan peningkatan kinerja.</p> <p><a href="#">Untuk informasi selengkapnya tentang Amazon MQ yang tersedia untuk versi RabbitMQ, lihat Amazon MQ untuk versi mesin RabbitMQ.</a></p>
1 Maret 2023	<p>Amazon MQ untuk RabbitMQ sekarang mendukung RabbitMQ versi 3.10.17.</p> <p>Untuk informasi selengkapnya tentang perbaikan dan fitur dalam rilis ini, lihat berikut ini:</p> <ul style="list-style-type: none"><li>• <a href="#">Catatan rilis RabbitMQ 3.10.17 pada repositori server RabbitMQ</a> GitHub</li><li>• <a href="#">RabbitMQ changelog</a></li></ul> <p>Untuk informasi selengkapnya tentang Amazon MQ yang didukung untuk versi RabbitMQ dan peningkatan broker, lihat. <a href="#">Mengelola Amazon MQ untuk versi mesin RabbitMQ</a></p>

Tanggal	Pembaruan Dokumentasi
21 Februari 2023	<p>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 <a href="#">Enkripsi saat istirahat</a>.</p> <p>Amazon MQ mendukung penggunaan AWS KMS kunci dengan cara berikut.</p> <ul style="list-style-type: none"><li>• Kunci KMS milik Amazon MQ (default) - Kunci dimiliki dan dikelola oleh Amazon MQ dan tidak ada di akun Anda.</li><li>• 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.</li><li>• Pilih kunci KMS yang ada dan dikelola pelanggan — Kunci KMS yang dikelola pelanggan dibuat dan dikelola oleh Anda di AWS Key Management Service (KMS).</li></ul>
13 Januari 2023	<p>Amazon MQ untuk RabbitMQ sekarang mendukung RabbitMQ versi 3.8.34.</p> <p>Untuk informasi selengkapnya tentang perbaikan dan fitur dalam rilis ini, lihat berikut ini:</p> <ul style="list-style-type: none"><li>• <a href="#">Catatan rilis RabbitMQ 3.8.34 pada repositori server RabbitMQ</a> GitHub</li><li>• <a href="#">RabbitMQ changelog</a></li></ul> <p>Untuk informasi selengkapnya tentang Amazon MQ yang didukung untuk versi RabbitMQ dan peningkatan broker, lihat. <a href="#">Mengelola Amazon MQ untuk versi mesin RabbitMQ</a></p>

Tanggal	Pembaruan Dokumentasi
Desember 15, 2022	<p>Amazon MQ untuk RabbitMQ sekarang mendukung RabbitMQ versi 3.9.24.</p> <p>Untuk informasi selengkapnya tentang perbaikan dan fitur dalam rilis ini, lihat berikut ini:</p> <ul style="list-style-type: none"><li>• <a href="#">Catatan rilis RabbitMQ 3.9.24 pada repositori server RabbitMQ</a> GitHub</li><li>• <a href="#">RabbitMQ changelog</a></li></ul> <p>Untuk informasi selengkapnya tentang Amazon MQ yang didukung untuk versi RabbitMQ dan peningkatan broker, lihat. <a href="#">Mengelola Amazon MQ untuk versi mesin RabbitMQ</a></p>
13 Desember 2022	<p>Amazon MQ sekarang tersedia di Wilayah Timur Tengah (UEA). Untuk informasi tentang wilayah yang tersedia, lihat <a href="#">AWS Wilayah dan Titik Akhir</a> dalam panduan Referensi AWS Umum.</p>
November 14, 2022	<p>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 hal berikut:</p> <ul style="list-style-type: none"><li>• <a href="#">Catatan rilis RabbitMQ 3.10.10</a></li><li>• <a href="#">RabbitMQ changelog</a></li></ul> <p>Untuk informasi selengkapnya tentang Amazon MQ yang didukung untuk versi RabbitMQ dan peningkatan broker, lihat. <a href="#">Mengelola Amazon MQ untuk versi mesin RabbitMQ</a></p>
9 November 2022	<p>Amazon MQ sekarang mendukung ActiveMQ 5.17.2, rilis versi mesin minor baru. Untuk informasi selengkapnya, lihat hal berikut:</p> <ul style="list-style-type: none"><li>• <a href="#">ActiveMQ 5.17.2 Halaman Rilis</a></li><li>• <a href="#">Mengelola versi mesin Amazon MQ for ActiveMQ</a></li><li>• <a href="#">Meningkatkan versi mesin broker Amazon MQ</a></li><li>• <a href="#">Bekerja dengan file konfigurasi Spring XML</a></li></ul>

Tanggal	Pembaruan Dokumentasi
17 Agustus 2022	<p>Amazon MQ sekarang mendukung ActiveMQ 5.17.1, rilis versi mesin utama baru. Untuk informasi selengkapnya, lihat hal berikut:</p> <ul style="list-style-type: none"><li>• <a href="#">ActiveMQ 5.17.1 Halaman Rilis</a></li><li>• <a href="#">Mengelola versi mesin Amazon MQ for ActiveMQ</a></li><li>• <a href="#">Meningkatkan versi mesin broker Amazon MQ</a></li><li>• <a href="#">Bekerja dengan file konfigurasi Spring XML</a></li></ul>
14 Juli 2022	<p>Amazon MQ sekarang mendukung ActiveMQ 5.16.5, rilis versi mesin minor. Untuk informasi selengkapnya, lihat hal berikut:</p> <ul style="list-style-type: none"><li>• <a href="#">ActiveMQ 5.16.5 Halaman Rilis</a></li><li>• <a href="#">Mengelola versi mesin Amazon MQ for ActiveMQ</a></li><li>• <a href="#">Bekerja dengan file konfigurasi Spring XML</a></li><li>• <a href="#">Meningkatkan versi mesin broker Amazon MQ</a></li></ul>
4 Mei, 2022	<p>Amazon MQ menambahkan bahasa inklusif untuk <code>networkConnector</code> elemen dalam konfigurasi broker.</p> <ul style="list-style-type: none"><li>• <a href="#">Membuat dan mengonfigurasi jaringan broker Amazon MQ</a></li></ul>
April 25, 2022	<p>Amazon MQ Rilis ini menambahkan status <code>CRITICAL_ACTION_REQUIRED</code> broker dan properti <code>ActionRequired</code> API. <code>CRITICAL_ACTION_REQUIRED</code> memberi tahu Anda ketika broker Anda terdegradasi. <code>ActionRequired</code> memberi Anda kode yang dapat Anda gunakan untuk menemukan petunjuk di Panduan Pengembang tentang cara mengatasi masalah.</p> <ul style="list-style-type: none"><li>• <a href="#">the section called “Pemecahan masalah: Kode yang diperlukan tindakan Amazon MQ”</a></li><li>• <a href="#">ActionRequired</a> dokumentasi di Referensi API Amazon MQ.</li></ul>

Tanggal	Pembaruan Dokumentasi
20 April 2022	<p>Amazon MQ sekarang mendukung ActiveMQ 5.16.4, rilis versi mesin minor. Untuk informasi selengkapnya, lihat hal berikut:</p> <ul style="list-style-type: none"><li>• <a href="#">ActiveMQ 5.16.4 Halaman Rilis</a></li><li>• <a href="#">Mengelola versi mesin Amazon MQ for ActiveMQ</a></li><li>• <a href="#">Bekerja dengan file konfigurasi Spring XML</a></li><li>• <a href="#">Meningkatkan versi mesin broker Amazon MQ</a></li></ul>
1 Maret 2022	<p>Amazon MQ sekarang tersedia di Wilayah Asia Pasifik (Jakarta). Untuk informasi tentang wilayah yang tersedia, lihat <a href="#">AWS Wilayah dan Titik Akhir</a> dalam panduan Referensi AWS Umum.</p>
25 Februari 2022	<p>Amazon MQ untuk RabbitMQ sekarang mendukung RabbitMQ versi 3.8.27. Untuk informasi selengkapnya tentang perbaikan dan fitur dalam rilis ini, lihat berikut ini:</p> <ul style="list-style-type: none"><li>• <a href="#">Catatan rilis RabbitMQ 3.8.27 pada repositori server RabbitMQ</a> GitHub</li><li>• <a href="#">RabbitMQ changelog</a></li></ul> <p>Untuk informasi selengkapnya tentang Amazon MQ yang didukung untuk versi RabbitMQ dan peningkatan broker, lihat <a href="#">Mengelola Amazon MQ untuk versi mesin RabbitMQ</a></p>
16 Februari 2022	<p>Amazon MQ sekarang tersedia di Wilayah Afrika (Cape Town). Untuk informasi tentang wilayah yang tersedia, lihat <a href="#">AWS Wilayah dan Titik Akhir</a> dalam panduan Referensi AWS Umum.</p>

Tanggal	Pembaruan Dokumentasi
14 Februari 2022	<p>Amazon MQ untuk RabbitMQ sekarang mendukung RabbitMQ versi 3.9.13. <a href="#">Upgrade versi minor otomatis</a> tidak dapat digunakan untuk meningkatkan dari Rabbit 3.8 ke 3.9. Untuk melakukannya, <a href="#">tingkatkan broker Anda secara manual</a>.</p> <p>Untuk informasi lebih lanjut tentang fitur baru yang diperkenalkan di RabbitMQ 3.9, lihat <a href="#">halaman catatan rilis untuk versi 3.9.0</a> di situs web. GitHub</p> <div data-bbox="402 575 1507 793"><p> Note</p><p>Saat ini, Amazon MQ tidak mendukung <a href="#">aliran</a>, atau menggunakan logging terstruktur di JSON, diperkenalkan di RabbitMQ 3.9.</p></div> <p>Untuk informasi selengkapnya tentang perbaikan dan fitur dalam rilis ini, lihat berikut ini:</p> <ul style="list-style-type: none"><li>• <a href="#">Catatan rilis RabbitMQ 3.9.13 pada repositori server RabbitMQ</a> GitHub</li><li>• <a href="#">RabbitMQ changelog</a></li></ul> <p>Untuk informasi selengkapnya tentang Amazon MQ yang didukung untuk versi RabbitMQ dan peningkatan broker, lihat. <a href="#">Mengelola Amazon MQ untuk versi mesin RabbitMQ</a></p>
Februari 07, 2022	<p>Amazon MQ untuk RabbitMQ memperkenalkan metrik broker baru, memungkinkan Anda memantau pemanfaatan sumber daya rata-rata di ketiga node dalam penerapan cluster.</p> <p>Untuk informasi selengkapnya, lihat hal berikut:</p> <ul style="list-style-type: none"><li>• <a href="#">the section called “Pencatatan dan pemantauan Amazon MQ untuk broker RabbitMQ”</a></li></ul>




Tanggal	Pembaruan Dokumentasi
18 Januari 2022	<p>Amazon MQ untuk RabbitMQ sekarang mendukung RabbitMQ versi 3.8.26.</p> <p>Untuk informasi selengkapnya tentang perbaikan dan fitur dalam rilis ini, lihat berikut ini:</p> <ul style="list-style-type: none"><li>• <a href="#">Catatan rilis RabbitMQ 3.8.26 pada repositori server RabbitMQ</a> GitHub</li><li>• <a href="#">RabbitMQ changelog</a></li></ul> <p>Untuk informasi selengkapnya tentang Amazon MQ yang didukung untuk versi RabbitMQ dan peningkatan broker, lihat. <a href="#">Mengelola Amazon MQ untuk versi mesin RabbitMQ</a></p>
Januari 13, 2022	<p>Amazon MQ memperkenalkan kode <code>RABBITMQ_MEMORY_ALARM</code> 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, menyelesaikan, dan mencegah alarm memori tinggi. Untuk informasi selengkapnya, lihat hal berikut.</p> <ul style="list-style-type: none"><li>• <a href="#">the section called “ RABBITMQ_MEMORY_ALARM ”</a></li></ul>
6 Januari 2022	<p>Saat Anda mengonfigurasi CloudWatch Log untuk Amazon MQ untuk broker ActiveMQ, Amazon MQ mendukung penggunaan <a href="#">aws:SourceAccount</a> dan kunci konteks kondisi global dalam kebijakan berbasis <a href="#">aws:SourceArn</a> sumber daya IAM untuk mencegah masalah deputi yang membingungkan. Untuk informasi selengkapnya, lihat hal berikut.</p> <ul style="list-style-type: none"><li>• <a href="#">the section called “Pencegahan wakil bingung lintas layanan”</a></li></ul>

Tanggal	Pembaruan Dokumentasi
Desember 20, 2021	<p><a href="#">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 transportasi yang didukung, serta metrik baru tambahan yang memungkinkan Anda memantau jumlah node yang terhubung ke broker Anda di jaringan broker.</a> Untuk informasi selengkapnya, lihat hal berikut.</p> <ul style="list-style-type: none"><li>• <a href="#">the section called “Pencatatan dan pemantauan Amazon MQ untuk broker ActiveMQ”</a></li></ul>
November 16, 2021	<p>Amazon MQ untuk RabbitMQ sekarang mendukung RabbitMQ versi 3.8.23. Untuk informasi selengkapnya tentang perbaikan dan fitur dalam rilis ini, lihat berikut ini:</p> <ul style="list-style-type: none"><li>• <a href="#">Catatan rilis RabbitMQ 3.8.23 pada repositori server RabbitMQ GitHub</a></li><li>• <a href="#">RabbitMQ changelog</a></li></ul> <p>Untuk informasi selengkapnya tentang Amazon MQ yang didukung untuk versi RabbitMQ dan peningkatan broker, lihat. <a href="#">Mengelola Amazon MQ untuk versi mesin RabbitMQ</a></p>
Oktober 12, 2021	<p>Amazon MQ sekarang mendukung ActiveMQ 5.16.3, rilis versi mesin minor. Untuk informasi selengkapnya, lihat hal berikut:</p> <ul style="list-style-type: none"><li>• <a href="#">ActiveMQ 5.16.3 Halaman Rilis</a></li><li>• <a href="#">Mengelola versi mesin Amazon MQ for ActiveMQ</a></li><li>• <a href="#">Meningkatkan versi mesin broker Amazon MQ</a></li><li>• <a href="#">Bekerja dengan file konfigurasi Spring XML</a></li></ul>

Tanggal	Pembaruan Dokumentasi
8 September 2021	<p>Amazon MQ untuk RabbitMQ sekarang mendukung RabbitMQ versi 3.8.22.</p> <p>Rilis ini mencakup perbaikan untuk masalah dengan antrian menggunakan <a href="#">TTL per pesan (waktu untuk hidup)</a>, diidentifikasi dalam versi yang didukung sebelumnya, RabbitMQ 3.8.17. Kami merekomendasikan untuk meningkatkan broker Anda yang ada ke versi 3.8.22.</p> <p>Untuk informasi selengkapnya tentang perbaikan dan fitur dalam rilis ini, lihat berikut ini:</p> <ul style="list-style-type: none"><li>• <a href="#">Catatan rilis RabbitMQ 3.8.22 pada repositori server RabbitMQ</a> GitHub</li><li>• <a href="#">RabbitMQ changelog</a></li></ul> <p>Untuk informasi selengkapnya tentang Amazon MQ yang didukung untuk versi RabbitMQ dan peningkatan broker, lihat <a href="#">Mengelola Amazon MQ untuk versi mesin RabbitMQ</a></p>
25 Agustus 2021	<p><a href="#">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</a> Kami merekomendasikan menggunakan versi 3.8.11.</p>
29 Juli 2021	<p>Amazon MQ untuk RabbitMQ sekarang mendukung RabbitMQ versi 3.8.17. Untuk informasi selengkapnya tentang perbaikan dan fitur yang ada dalam pembaruan ini, lihat:</p> <ul style="list-style-type: none"><li>• <a href="#">Catatan rilis RabbitMQ 3.8.17 pada repositori server RabbitMQ</a> GitHub</li><li>• <a href="#">RabbitMQ changelog</a></li><li>• <a href="#">Mengelola Amazon MQ untuk versi mesin RabbitMQ</a></li></ul>
16 Juli 2021	<p>Anda sekarang dapat menyesuaikan jendela pemeliharaan broker Amazon MQ menggunakan AWS Management Console, AWS CLI, atau Amazon MQ API. Untuk mempelajari lebih lanjut tentang jendela pemeliharaan broker, lihat yang berikut ini.</p> <ul style="list-style-type: none"><li>• <a href="#">Mempertahankan broker Amazon MQ</a></li></ul>

Tanggal	Pembaruan Dokumentasi
6 Juli 2021	<p>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 hal berikut:</p> <ul style="list-style-type: none"><li>• <a href="#">Plugin pertukaran Hash yang konsisten</a></li><li>• <a href="#">Jenis Pertukaran Hash Konsisten RabbitMQ</a> pada repositori RabbitMQ GitHub</li></ul>
7 Juni 2021	<p>Amazon MQ kini mendukung ActiveMQ 5.16.2, rilis baru versi mesin utama. Untuk informasi lebih lanjut, lihat hal berikut:</p> <ul style="list-style-type: none"><li>• <a href="#">ActiveMQ 5.16.2 Halaman Rilis</a></li><li>• <a href="#">Mengelola versi mesin Amazon MQ for ActiveMQ</a></li><li>• <a href="#">Meningkatkan versi mesin broker Amazon MQ</a></li><li>• <a href="#">Bekerja dengan file konfigurasi Spring XML</a></li></ul>
26 Mei 2021	<p>Amazon MQ untuk RabbitMQ kini tersedia di Wilayah China (Beijing) dan China (Ningxia). Untuk informasi tentang wilayah yang tersedia, lihat <a href="#">Wilayah dan Titik Akhir AWS</a>.</p>
18 Mei 2021	<p>Amazon MQ for RabbitMQ mengimplementasikan broker default.</p> <p>Ketika Anda pertama kali membuat broker, Amazon MQ membuat serangkaian kebijakan broker dan batas vhost berdasarkan tipe instans dan mode deployment yang Anda pilih, untuk mengoptimalkan performa broker. Untuk informasi lebih lanjut, lihat hal berikut:</p> <ul style="list-style-type: none"><li>• <a href="#">Amazon MQ untuk default broker RabbitMQ</a></li></ul>
5 Mei 2021	<p>Amazon MQ kini mendukung ActiveMQ 5.15.15. Untuk informasi lebih lanjut, lihat hal berikut:</p> <ul style="list-style-type: none"><li>• <a href="#">ActiveMQ 5.15.15 Halaman Rilis</a></li><li>• <a href="#">Mengelola versi mesin Amazon MQ for ActiveMQ</a></li><li>• <a href="#">Bekerja dengan file konfigurasi Spring XML</a></li></ul>


Tanggal	Pembaruan Dokumentasi
5 Mei 2021	<p>Amazon MQ mulai melacak perubahan pada kebijakan AWS terkelola. Untuk informasi selengkapnya, lihat hal berikut:</p> <ul style="list-style-type: none"><li>• <a href="#">the section called “Kebijakan terkelola AWS”</a></li></ul>
14 April 2021	<p>Amazon MQ kini tersedia di Wilayah China (Beijing) dan China (Ningxia). Untuk informasi tentang wilayah yang tersedia, lihat <a href="#">Wilayah dan Titik Akhir AWS</a>.</p>
7 April 2021	<p>Amazon MQ kini mendukung RabbitMQ 3.8.11. Untuk informasi selengkapnya tentang perbaikan dan fitur yang ada dalam pembaruan ini, lihat:</p> <ul style="list-style-type: none"><li>• <a href="#">Catatan rilis RabbitMQ 3.8.11 pada repositori server RabbitMQ</a> GitHub</li><li>• <a href="#">RabbitMQ changelog</a></li><li>• <a href="#">Mengelola Amazon MQ untuk versi mesin RabbitMQ</a></li></ul>
1 April 2021	<p>Amazon MQ kini tersedia di Wilayah Asia Pacific (Osaka). Untuk informasi tentang wilayah yang tersedia, lihat <a href="#">Wilayah dan titik akhir Amazon MQ</a>.</p>
21 Desember 2020	<p>Amazon MQ kini mendukung ActiveMQ 5.15.14. Untuk informasi lebih lanjut, lihat hal berikut:</p> <ul style="list-style-type: none"><li>• <a href="#">Catatan Rilis ActiveMQ 5.15.14</a></li><li>• <a href="#">Mengelola versi mesin Amazon MQ for ActiveMQ</a></li><li>• <a href="#">Bekerja dengan file konfigurasi Spring XML</a></li><li>• <div data-bbox="435 1360 1507 1675" style="border: 1px solid #f08080; border-radius: 10px; padding: 10px;"><p> <b>Important</b></p><p>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 <a href="#">AMQ-8104</a>.</p></div></li></ul>

Tanggal	Pembaruan Dokumentasi
4 November 2020	<p>Amazon MQ kini mendukung <a href="#">RabbitMQ</a>, broker pesan sumber terbuka populer. Ini memungkinkan Anda untuk memigrasikan broker pesan RabbitMQ yang ada ke AWS tanpa harus menulis ulang kode.</p> <p>Amazon MQ for RabbitMQ mengelola broker pesan individu dan terkluster serta menangani tugas seperti penyediaan infrastruktur, penyiapan broker, dan pembaruan perangkat lunak.</p> <ul style="list-style-type: none"><li>• Amazon MQ mendukung RabbitMQ 3.8.6. Untuk informasi selengkapnya tentang versi mesin yang didukung, lihat <a href="#">the section called “Manajemen versi”</a>.</li><li>• <a href="#">Tingkat Gratis AWS</a> menyertakan hingga 750 jam dari broker <code>mq.t3.micro</code> instans tunggal dan penyimpanan hingga 20 GB per bulan selama satu tahun. Untuk informasi selengkapnya tentang tipe instans yang didukung, lihat <a href="#">Broker instance types</a>.</li><li>• <a href="#">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.</a> Untuk informasi selengkapnya tentang protokol dan cipher suite yang didukung, lihat <a href="#">the section called “Protokol Amazon MQ for RabbitMQ”</a>.</li><li>• Amazon MQ for RabbitMQ tersedia di semua wilayah tempat Amazon MQ tersedia saat ini. Untuk mempelajari selengkapnya tentang semua wilayah yang tersedia, lihat <a href="#">Tabel Wilayah AWS</a>.</li></ul> <p>Untuk mulai menggunakan Amazon MQ, membuat broker, dan menghubungkan aplikasi berbasis JVM ke broker RabbitMQ Anda, lihat <a href="#">the section called “Membuat dan menghubungkan ke broker RabbitMQ”</a>.</p>
22 Oktober 2020	<p>Amazon MQ mendukung ActiveMQ 5.15.13. Untuk informasi lebih lanjut, lihat hal berikut:</p> <ul style="list-style-type: none"><li>• <a href="#">Catatan Rilis ActiveMQ 5.15.13</a></li><li>• <a href="#">Mengelola versi mesin Amazon MQ for ActiveMQ</a></li><li>• <a href="#">Bekerja dengan file konfigurasi Spring XML</a></li></ul>

Tanggal	Pembaruan Dokumentasi
30 September 2020	Amazon MQ kini tersedia di Wilayah Europe (Milan). Untuk informasi tentang wilayah yang tersedia, lihat <a href="#">Wilayah dan titik akhir Amazon MQ</a> .
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 antrian. Untuk informasi selengkapnya, lihat <a href="#">Mengintegrasikan LDAP dengan ActiveMQ</a> .
17 Juli 2020	Amazon MQ kini mendukung tipe instans <code>mq.t3.micro</code> . Untuk informasi selengkapnya, lihat <a href="#">Broker instance types</a> .
30 Juni 2020	<p>Amazon MQ mendukung ActiveMQ 5.15.12. Untuk informasi lebih lanjut, lihat hal berikut:</p> <ul style="list-style-type: none"> <li>• <a href="#">Catatan Rilis ActiveMQ 5.15.12</a></li> <li>• <a href="#">Mengelola versi mesin Amazon MQ for ActiveMQ</a></li> <li>• <a href="#">Bekerja dengan file konfigurasi Spring XML</a></li> </ul>
30 April 2020	<p>Amazon MQ mendukung elemen pengumpulan anak baru, <code>systemUsage</code>, pada elemen <code>broker</code>. Untuk informasi selengkapnya, lihat <a href="#">systemUsage</a>.</p> <p>Amazon MQ juga mendukung tiga atribut baru pada elemen anak <code>kahaDB</code>.</p> <ul style="list-style-type: none"> <li>• <code>journalDiskSyncInterval</code> - Interval (mdtk) untuk kapan harus melakukan sinkronisasi disk jika <code>journalDiskSyncStrategy=periodic</code>.</li> <li>• <code>journalDiskSyncStrategy</code> - mengonfigurasi kebijakan sinkronisasi disk.</li> <li>• <code>preallocationStrategy</code> - mengonfigurasi cara broker akan melakukan pra-alokasi file jurnal ketika file jurnal baru diperlukan.</li> </ul> <p>Untuk informasi selengkapnya, lihat <a href="#">Atribut</a>.</p>

Tanggal	Pembaruan Dokumentasi
3 Maret 2020	<p>Amazon MQ mendukung dua metrik baru CloudWatch</p> <ul style="list-style-type: none"><li>• <code>TempPercentUsage</code> - Persentase penyimpanan sementara yang tersedia dan digunakan oleh pesan tidak tetap.</li><li>• <code>JobSchedulerStorePercentUsage</code> - Persentase ruang disk yang digunakan oleh penyimpanan penjadwal tugas.</li></ul> <p>Untuk informasi selengkapnya, lihat <a href="#">Monitoring Amazon MQ using CloudWatch</a>.</p>
4 Februari 2020	<p>Amazon MQ tersedia di wilayah Asia Pacific (Hong Kong) dan Middle East (Bahrain). Untuk informasi tentang wilayah yang tersedia, lihat <a href="#">Wilayah dan Titik Akhir AWS</a>.</p>
22 Januari 2020	<p>Amazon MQ mendukung ActiveMQ 5.15.10. Untuk informasi lebih lanjut, lihat hal berikut:</p> <ul style="list-style-type: none"><li>• <a href="#">Catatan Rilis ActiveMQ 5.15.10</a></li><li>• <a href="#">Mengelola versi mesin Amazon MQ for ActiveMQ</a></li><li>• <a href="#">Bekerja dengan file konfigurasi Spring XML</a></li></ul>
19 Desember 2019	<p>Amazon MQ tersedia di wilayah Europe (Stockholm) dan South America (São Paulo). Untuk informasi tentang wilayah yang tersedia, lihat <a href="#">Wilayah dan Titik Akhir AWS</a>.</p>




Tanggal	Pembaruan Dokumentasi
16 Desember 2019	<p>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 Availability Zone, gunakan Amazon EFS. Untuk memanfaatkan latensi rendah dan throughput yang tinggi, gunakan Amazon EBS.</p> <div data-bbox="402 541 1507 991" style="border: 1px solid #f08080; border-radius: 10px; padding: 10px;"><p> <b>Important</b></p><ul style="list-style-type: none"><li>• Anda dapat menggunakan Amazon EBS hanya dengan keluarga tipe instans broker mq.m5.</li><li>• Meski Anda dapat mengubah tipe instans broker, Anda tidak dapat mengubah tipe penyimpanan broker setelah Anda membuat broker.</li><li>• Amazon EBS mereplikasi data dalam satu Availability Zone dan tidak mendukung mode deployment <a href="#">ActiveMQ aktif/siaga</a>.</li></ul></div> <p>Untuk informasi lebih lanjut, lihat hal berikut:</p> <ul style="list-style-type: none"><li>• <a href="#">Storage</a></li><li>• <a href="#">Pilih jenis penyimpanan broker yang tepat untuk throughput terbaik</a></li><li>• Properti <code>storageType</code> dari sumber daya <a href="#">broker-instance-options</a> dalam Referensi REST API Amazon MQ</li><li>• Metrik <code>BurstBalance</code>, <code>VolumeReadOps</code>, dan <code>VolumeWriteOps</code> di bagian <a href="#">Metrik Amazon MQ untuk ActiveMQ</a>.</li></ul>
18 Oktober 2019	<p>Dua CloudWatch metrik Amazon tersedia: <code>TotalEnqueueCount</code> dan <code>TotalDequeueCount</code>. Untuk informasi selengkapnya, lihat <a href="#">Metrik tujuan ActiveMQ (antrean dan topik)</a>.</p>

Tanggal	Pembaruan Dokumentasi
11 Oktober 2019	<p>Amazon MQ kini mendukung titik akhir kepatuhan Standar Pemrosesan Informasi Federal 140-2 (FIPS) di wilayah komersial AS.</p> <p>Untuk informasi lebih lanjut, lihat hal berikut:</p> <ul style="list-style-type: none"> <li>• <a href="#">Standar Pemrosesan Informasi Federal (FIPS) 140-2</a></li> <li>• <a href="#">Wilayah dan Titik Akhir Amazon MQ</a></li> </ul>
30 September 2019	<p>Amazon MQ kini menyertakan kemampuan untuk menskalakan broker Anda dengan mengubah tipe instans host. Untuk informasi selengkapnya, lihat properti <code>hostInstanceType</code> dari <a href="#">UpdateBrokerInput</a>, dan properti <code>pendingHostInstanceType</code> dari <a href="#">DescribeBrokerOutput</a>.</p>
30 Agustus 2019	<p>Kini Anda dapat memperbarui grup keamanan yang terkait dengan broker, baik di konsol maupun dengan <a href="#">UpdateBrokerInput</a>.</p>
22 Juli 2019	<p>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 <a href="#">Enkripsi saat istirahat</a>.</p> <p>Amazon MQ mendukung penggunaan AWS KMS kunci dengan cara berikut.</p> <ul style="list-style-type: none"> <li>• AWS kunci KMS yang dimiliki - Kuncinya dimiliki Amazon MQ dan tidak ada di akun Anda.</li> <li>• AWS kunci KMS AWS terkelola - Kunci KMS terkelola (<code>aws/mq</code>) adalah kunci KMS di akun Anda yang dibuat, dikelola, dan digunakan atas nama Anda oleh Amazon MQ.</li> <li>• Pilih kunci CMK yang ada dan dikelola pelanggan — Kunci CMK yang dikelola pelanggan dibuat dan dikelola oleh Anda di AWS Key Management Service (KMS).</li> </ul>
19 Juni 2019	<p>Amazon MQ tersedia di wilayah Europe (Paris) dan Asia Pacific (Mumbai). Untuk informasi tentang wilayah yang tersedia, lihat <a href="#">Wilayah dan Titik Akhir AWS</a>.</p>

Tanggal	Pembaruan Dokumentasi
12 Juni 2019	Amazon MQ tersedia di wilayah Canada (Central). Untuk informasi tentang wilayah yang tersedia, lihat <a href="#">Wilayah dan Titik Akhir AWS</a> .
3 Juni 2019	Dua CloudWatch metrik Amazon baru tersedia: <code>EstablishedConnectionsCount</code> dan <code>InactiveDurableSubscribers</code> . Untuk informasi selengkapnya, lihat hal berikut: <ul style="list-style-type: none"><li>• <a href="#">Monitoring Amazon MQ using CloudWatch</a></li><li>• <a href="#">Metrik Amazon MQ untuk ActiveMQ</a></li></ul>
10 Mei 2019	Penyimpanan data untuk tipe instans <code>mq.t2.micro</code> terbatas pada 20 GB. Untuk informasi lebih lanjut, lihat hal berikut: <ul style="list-style-type: none"><li>• <a href="#">the section called “Penyimpanan Data”</a></li><li>• <a href="#">Broker instance types</a></li></ul>
29 April 2019	Kini Anda dapat menggunakan kebijakan berbasis tanda dan izin tingkat sumber daya. Untuk informasi lebih lanjut, lihat hal berikut: <ul style="list-style-type: none"><li>• <a href="#">Cara kerja Amazon MQ dengan IAM</a></li><li>• <a href="#">Izin tingkat sumber daya untuk tindakan API Amazon MQ</a></li></ul>
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: <ul style="list-style-type: none"><li>• <a href="#">Opsi instans broker</a></li><li>• <a href="#">Jenis mesin broker</a></li></ul>
8 April 2019	Amazon MQ mendukung ActiveMQ 5.15.9. Untuk informasi lebih lanjut, lihat hal berikut: <ul style="list-style-type: none"><li>• <a href="#">Catatan Rilis ActiveMQ 5.15.9</a></li><li>• <a href="#">Mengelola versi mesin Amazon MQ for ActiveMQ</a></li><li>• <a href="#">Bekerja dengan file konfigurasi Spring XML</a></li></ul>

Tanggal	Pembaruan Dokumentasi
4 Maret 2019	<p>Peningkatan dokumentasi untuk mengonfigurasi failover dinamis dan menyeimbangkan ulang klien untuk jaringan broker. Aktifkan failover dinamis dengan mengonfigurasi <code>transportConnectors</code> bersama dengan opsi konfigurasi <code>networkConnectors</code> . Untuk informasi lebih lanjut, lihat hal berikut:</p> <ul style="list-style-type: none"><li>• <a href="#">Failover Dinamis dengan Konektor Transportasi</a></li><li>• <a href="#">Jaringan broker Amazon MQ</a></li><li>• <a href="#">Amazon MQ Broker Configuration Parameters</a></li></ul>
27 Februari 2019	<p>Amazon MQ tersedia di Wilayah Europe (London) selain wilayah berikut:</p> <ul style="list-style-type: none"><li>• Asia Pacific (Singapore)</li><li>• US East (Ohio)</li><li>• AS Timur (Virginia Utara)</li><li>• AS Barat (California Utara)</li><li>• US West (Oregon)</li><li>• Asia Pacific (Tokyo)</li><li>• Asia Pacific (Seoul)</li><li>• Asia Pacific (Sydney)</li><li>• Europe (Frankfurt)</li><li>• Europe (Ireland)</li></ul>
24 Januari 2019	<p>Konfigurasi default kini menyertakan kebijakan untuk membersihkan tujuan yang tidak aktif.</p>
17 Januari 2019	<p>Tipe instans <code>mq.t2.micro</code> Amazon MQ kini hanya mendukung 100 koneksi per protokol tingkat wire. Untuk informasi selengkapnya, lihat, <a href="#">Quotas in Amazon MQ</a>.</p>



Tanggal	Pembaruan Dokumentasi
19 Desember 2018	<p>Anda dapat mengonfigurasi serangkaian broker Amazon MQ dalam jaringan broker. Untuk informasi selengkapnya, lihat bagian berikut:</p> <ul style="list-style-type: none"><li>• <a href="#">Jaringan broker Amazon MQ</a></li><li>• <a href="#">Creating and Configuring a Network of Brokers</a></li><li>• <a href="#">Mengonfigurasi Jaringan Broker dengan Benar</a></li><li>• <a href="#">networkConnector</a></li><li>• <a href="#">networkConnectionStartasinkron</a></li></ul>
11 Desember 2018	<p>Amazon MQ mendukung ActiveMQ 5.15.8, 5.15.6, dan 5.15.0.</p> <ul style="list-style-type: none"><li>• Bug yang diselesaikan dan peningkatan di ActiveMQ:<ul style="list-style-type: none"><li>• <a href="#">Catatan Rilis ActiveMQ 5.15.8</a></li><li>• <a href="#">Catatan Rilis ActiveMQ 5.15.7</a></li></ul></li></ul>
5 Desember 2018	<p>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 <a href="#">Menandai sumber daya</a>.</p>
19 November 2018	<p>AWS telah memperluas program kepatuhan SOC untuk memasukkan Amazon MQ sebagai layanan yang sesuai <a href="#">dengan SOC</a>.</p>
15 Oktober 2018	<ul style="list-style-type: none"><li>• Jumlah maksimum grup per pengguna adalah 20. Untuk informasi selengkapnya, lihat <a href="#">Pengguna</a>.</li><li>• Jumlah maksimum koneksi per broker, per protokol tingkat wire adalah 1.000. Untuk informasi selengkapnya, lihat <a href="#">Pialang</a>.</li></ul>
2 Oktober 2018	<p>AWS telah memperluas program kepatuhan HIPAA untuk memasukkan Amazon MQ sebagai Layanan yang Memenuhi Syarat <a href="#">HIPAA</a>.</p>



Tanggal	Pembaruan Dokumentasi
27 September 2018	<p>Amazon MQ mendukung ActiveMQ 5.15.6, selain 5.15.0. Untuk informasi lebih lanjut, lihat hal berikut:</p> <ul style="list-style-type: none"><li>• <a href="#">Mengedit versi mesin broker, jenis instans, CloudWatchlog, dan preferensi pemeliharaan</a></li><li>• Bug yang diselesaikan dan peningkatan dalam dokumentasi ActiveMQ:<ul style="list-style-type: none"><li>• <a href="#">Catatan Rilis ActiveMQ 5.15.6</a></li><li>• <a href="#">Catatan Rilis ActiveMQ 5.15.5</a></li><li>• <a href="#">Catatan Rilis ActiveMQ 5.15.4</a></li><li>• <a href="#">Catatan Rilis ActiveMQ 5.15.3</a></li><li>• <a href="#">Catatan Rilis ActiveMQ 5.15.2</a></li><li>• <a href="#">Catatan Rilis ActiveMQ 5.15.1</a></li></ul></li><li>• <a href="#">Klien ActiveMQ 5.15.6</a></li></ul>
31 Agustus 2018	<ul style="list-style-type: none"><li>• Metrik berikut tersedia:<ul style="list-style-type: none"><li>• <code>CurrentConnectionsCount</code></li><li>• <code>TotalConsumerCount</code></li><li>• <code>TotalProducerCount</code></li></ul></li></ul> <p>Untuk informasi selengkapnya, lihat bagian <a href="#">Metrik Amazon MQ untuk ActiveMQ</a>.</p> <ul style="list-style-type: none"><li>• Alamat IP broker ditampilkan pada halaman Detail.</li></ul> <div data-bbox="431 1373 1508 1591" style="border: 1px solid #0070C0; border-radius: 10px; padding: 10px;"><p> <b>Note</b></p><p>Untuk broker dengan aksesibilitas publik yang dinonaktifkan, alamat IP internal ditampilkan.</p></div>

Tanggal	Pembaruan Dokumentasi
30 Agustus 2018	<p>Amazon MQ tersedia di Wilayah Asia Pacific (Singapore) selain wilayah berikut:</p> <ul style="list-style-type: none"><li>• US East (Ohio)</li><li>• AS Timur (Virginia Utara)</li><li>• AS Barat (California Utara)</li><li>• US West (Oregon)</li><li>• Asia Pacific (Tokyo)</li><li>• Asia Pacific (Seoul)</li><li>• Asia Pacific (Sydney)</li><li>• Europe (Frankfurt)</li><li>• Europe (Ireland)</li></ul>
30 Juli 2018	<p>Anda dapat mengonfigurasi Amazon MQ untuk mempublikasikan log umum dan audit ke Amazon CloudWatch Logs. Untuk informasi selengkapnya, lihat <a href="#">Configuring Amazon MQ to publish logs to Amazon CloudWatch Logs</a>.</p>
25 Juli 2018	<p>Amazon MQ tersedia di Wilayah Asia Pacific (Tokyo) dan Asia Pacific (Seoul) selain wilayah berikut:</p> <ul style="list-style-type: none"><li>• US East (Ohio)</li><li>• AS Timur (Virginia Utara)</li><li>• AS Barat (California Utara)</li><li>• US West (Oregon)</li><li>• Asia Pacific (Sydney)</li><li>• Europe (Frankfurt)</li><li>• Europe (Ireland)</li></ul>
19 Juli 2018	<p>Anda dapat menggunakan AWS CloudTrail untuk mencatat panggilan API Amazon MQ. Untuk informasi selengkapnya, lihat <a href="#">Logging Amazon MQ API calls using CloudTrail</a>.</p>

Tanggal	Pembaruan Dokumentasi
29 Juni 2018	<p>Selain <code>mq.t2.micro</code> dan <code>mq.m4.large</code>, tipe instans broker berikut tersedia untuk pengembangan, pengujian, dan beban kerja produksi reguler yang memerlukan throughput tinggi:</p> <ul style="list-style-type: none"><li>• <code>mq.m5.large</code></li><li>• <code>mq.m5.xlarge</code></li><li>• <code>mq.m5.2xlarge</code></li><li>• <code>mq.m5.4xlarge</code></li></ul> <p>Untuk informasi selengkapnya, lihat <a href="#">Broker instance types</a>.</p>
27 Juni 2018	<p>Amazon MQ tersedia di Wilayah US West (N. California) selain wilayah berikut:</p> <ul style="list-style-type: none"><li>• US East (Ohio)</li><li>• AS Timur (Virginia Utara)</li><li>• US West (Oregon)</li><li>• Asia Pacific (Sydney)</li><li>• Europe (Frankfurt)</li><li>• Europe (Ireland)</li></ul>



Tanggal	Pembaruan Dokumentasi
14 Juni 2018	<ul style="list-style-type: none"> <li>• Anda dapat menggunakan <a href="#">AWS::Amazon MQ::Broker</a> AWS CloudFormation sumber daya untuk melakukan tindakan berikut:               <ul style="list-style-type: none"> <li>• Membuat broker.</li> <li>• Menambahkan perubahan konfigurasi atau memodifikasi pengguna untuk broker yang ditentukan.</li> <li>• Mengembalikan informasi tentang broker yang ditentukan.</li> <li>• Menghapus broker yang ditentukan.</li> </ul> </li> </ul> <div data-bbox="435 632 1507 894" style="border: 1px solid #add8e6; border-radius: 10px; padding: 10px; margin: 10px 0;"> <p> <b>Note</b></p> <p>Saat Anda mengubah properti apa pun dari jenis properti Broker <a href="#">Amazon MQ Broker ConfigurationId</a> atau <a href="#">Amazon MQ Broker User, broker</a> segera di-boot ulang.</p> </div> <ul style="list-style-type: none"> <li>• Anda dapat menggunakan <a href="#">AWS::Amazon MQ::Configuration</a> AWS CloudFormation sumber daya untuk melakukan tindakan berikut:               <ul style="list-style-type: none"> <li>• Membuat konfigurasi.</li> <li>• Memperbarui konfigurasi yang ditentukan.</li> <li>• Mengembalikan informasi tentang konfigurasi yang ditentukan.</li> </ul> </li> </ul> <div data-bbox="435 1209 1507 1430" style="border: 1px solid #add8e6; border-radius: 10px; padding: 10px; margin: 10px 0;"> <p> <b>Note</b></p> <p>Anda dapat menggunakan AWS CloudFormation untuk memodifikasi—tetapi tidak menghapus—konfigurasi Amazon MQ.</p> </div>
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 selengkapnya, lihat <a href="#">Pengguna</a> .
13 Maret 2018	Pembuatan broker berlangsung sekitar 15 menit. Untuk informasi selengkapnya, lihat <a href="#">Selesai membuat broker</a> .

Tanggal	Pembaruan Dokumentasi
1 Maret 2018	<ul style="list-style-type: none"> <li>• Anda dapat menggunakan <a href="#">penyimpanan dan pengiriman bersamaan</a> untuk Apache KahaDB menggunakan atribut <a href="#">concurrentStoreAndDispatchQueues</a>.</li> <li>• <a href="#">CpuCreditBalance CloudWatch Metrik</a> tersedia untuk jenis instans <code>mq.t2.micro</code> broker.</li> </ul>
10 Januari 2018	<p>Perubahan berikut memengaruhi <a href="#">konsol Amazon MQ</a>:</p> <ul style="list-style-type: none"> <li>• Dalam daftar broker, kolom pembuatan tersembunyi secara default. Untuk menyesuaikan ukuran halaman dan kolom, pilih .</li> <li>• Pada <b>MyBroker</b> 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 <a href="#">Mengaktifkan koneksi masuk</a> yang diperbarui.</li> </ul>
9 Januari 2018	<ul style="list-style-type: none"> <li>• Izin untuk ID operasi REST <a href="#">UpdateBroker</a> tercantum dengan benar sebagai <code>mq:UpdateBroker</code> pada konsol IAM.</li> <li>• Izin <code>mq:DescribeEngine</code> yang salah dihapus dari konsol IAM.</li> </ul>

Tanggal	Pembaruan Dokumentasi
28 November 2017	<p>Ini adalah rilis awal dari Amazon MQ dan Rilis Developer Amazon MQ.</p> <ul style="list-style-type: none"><li>• Amazon MQ tersedia di wilayah berikut:<ul style="list-style-type: none"><li>• US East (Ohio)</li><li>• AS Timur (Virginia Utara)</li><li>• US West (Oregon)</li><li>• Asia Pacific (Sydney)</li><li>• Europe (Frankfurt)</li><li>• Europe (Ireland)</li></ul></li></ul> <p>Penggunaan tipe instans <code>mq.t2.micro</code> tunduk pada <a href="#">kredit CPU dan performa dasar</a>—dengan kemampuan untuk burst di atas tingkat dasar (untuk informasi selengkapnya, lihat metrik <a href="#">CpuCreditBalance</a> ).</p> <p>Jika aplikasi Anda membutuhkan performa tetap, pertimbangkan untuk menggunakan tipe instans <code>mq.m5.large</code> .</p> <ul style="list-style-type: none"><li>• Anda dapat membuat broker <code>mq.m4.large</code> dan <code>mq.t2.micro</code> .</li></ul> <p>Penggunaan tipe instans <code>mq.t2.micro</code> tunduk pada <a href="#">kredit CPU dan performa dasar</a>—dengan kemampuan untuk burst di atas tingkat dasar (untuk informasi selengkapnya, lihat metrik <a href="#">CpuCreditBalance</a> ).</p> <p>Jika aplikasi Anda membutuhkan performa tetap, pertimbangkan untuk menggunakan tipe instans <code>mq.m5.large</code> .</p> <ul style="list-style-type: none"><li>• Anda dapat menggunakan mesin broker ActiveMQ 5.15.0.</li><li>• Anda juga dapat membuat dan mengelola broker secara terprogram menggunakan Amazon <a href="#">MQ REST</a> API dan SDK. AWS</li><li>• Anda dapat mengakses broker menggunakan <a href="#">bahasa pemrograman yang didukung ActiveMQ</a> dan dengan mengaktifkan TLS secara eksplisit untuk protokol berikut:<ul style="list-style-type: none"><li>• <a href="#">AMQP</a></li><li>• <a href="#">MQTT</a></li><li>• MQTT lebih <a href="#">WebSocket</a></li><li>• <a href="#">OpenWire</a></li></ul></li></ul>

Tanggal	Pembaruan Dokumentasi
	<ul style="list-style-type: none"> <li>• <a href="#">MENGINJAK</a></li> <li>• STOMP berakhir WebSocket</li> <li>• Anda dapat terhubung ke broker ActiveMQ menggunakan <a href="#">berbagai klien ActiveMQ</a>. Kami merekomendasikan penggunaan <a href="#">Klien ActiveMQ</a>. Untuk informasi selengkapnya, lihat <a href="#">Connecting a Java application to your broker</a>.</li> <li>• Broker Anda dapat mengirim dan menerima pesan dengan berbagai ukuran.</li> </ul>

## Riwayat Dokumen Amazon MQ

Tabel berikut mencantumkan perubahan pada Panduan Developer Amazon MQ. Untuk rilis dan peningkatan fitur Amazon MQ, lihat [Catatan rilis Amazon MQ](#).

Tanggal	Pembaruan Dokumentasi
22 Agustus 2022	<p>Membuat chapter induk terpisah untuk Amazon MQ untuk ActiveMQ dan Amazon MQ untuk mesin RabbitMQ. Bab induk ini sekarang berisi detail mesin, tutorial, dan praktik terbaik:</p> <ul style="list-style-type: none"> <li>• <a href="#">Working with Amazon MQ for ActiveMQ</a></li> <li>• <a href="#">Working with Amazon MQ for RabbitMQ</a></li> </ul>
Januari 13, 2022	<p>Menambahkan bagian pemecahan masalah baru yang mencantumkan kode status yang dikembalikan Amazon MQ ketika broker dalam keadaan tidak sehat, bersama dengan informasi terperinci tentang mendiagnosis, dan memulihkan broker.</p> <ul style="list-style-type: none"> <li>• <a href="#">the section called “Pemecahan masalah: Kode yang diperlukan tindakan Amazon MQ”</a></li> </ul>
November 8, 2021	<p>Menambahkan tutorial baru yang menjelaskan pengaturan klien Python Pika dengan Amazon MQ untuk broker RabbitMQ.</p>

Tanggal	Pembaruan Dokumentasi
Oktober 8, 2021	<p>Menambahkan topik pemecahan masalah berikut untuk Amazon MQ untuk ActiveMQ dan Amazon MQ untuk mesin broker RabbitMQ:</p> <ul style="list-style-type: none"><li>• <a href="#">the section called “Menggunakan Python Pika dengan Amazon MQ for RabbitMQ”</a></li></ul>
22 September 2021	<p>Menambahkan topik berikut untuk memecahkan masalah koneksi umum, dan masalah otorisasi dengan Amazon MQ untuk broker ActiveMQ:</p> <ul style="list-style-type: none"><li>• <a href="#">Beberapa klien tidak dapat terhubung</a></li><li>• <a href="#">the section called “Bagaimana cara mengaktifkan plugin di Amazon MQ untuk RabbitMQ?”</a></li><li>• <a href="#">the section called “Saya tidak dapat mengubah konfigurasi VPC Amazon untuk broker.”</a></li></ul>
Agustus 12, 2021	<p>Menambahkan bagian berikut untuk menjelaskan pemecahan masalah umum saat bekerja dengan broker Amazon MQ.</p> <ul style="list-style-type: none"><li>• <a href="#">Pemecahan Masalah</a></li></ul>
29 Juli 2021	<p>Menambahkan bagian berikut untuk menjelaskan Amazon MQ untuk manajemen versi RabbitMQ dan meningkatkan broker Amazon MQ ke versi mesin minor dan utama baru saat didukung.</p> <ul style="list-style-type: none"><li>• <a href="#">the section called “Manajemen versi”</a></li></ul>
21 Juli 2021	<p>Menambahkan bagian berikut untuk menjelaskan menghubungkan broker Amazon MQ AWS Lambda sebagai sumber acara.</p> <ul style="list-style-type: none"><li>• <a href="#">Connect your Amazon MQ for ActiveMQ broker to Lambda</a></li><li>• <a href="#">Connect your Amazon MQ for RabbitMQ broker to Lambda</a></li></ul>

Tanggal	Pembaruan Dokumentasi
16 Juli 2021	<p>Menambahkan bagian berikut untuk menjelaskan jendela pemeliharaan broker Amazon MQ, dan cara menyesuaikan jendela pemeliharaan menggunakan AWS Management Console, AWS CLI, atau Amazon MQ API.</p> <ul style="list-style-type: none"><li>• <a href="#">the section called “Mempertahankan broker”</a></li></ul>
7 Juni 2021	<p>Bagian berikut ditambahkan untuk menjelaskan manajemen versi Amazon MQ for ActiveMQ serta meningkatkan broker Amazon MQ ke versi mesin minor dan utama baru karena mereka didukung.</p> <ul style="list-style-type: none"><li>• <a href="#">the section called “Manajemen versi”</a></li><li>• <a href="#">the section called “Meningkatkan versi mesin”</a></li></ul>
18 Mei 2021	<p>Bagian berikut ditambahkan untuk menjelaskan broker default Amazon MQ for RabbitMQ</p> <ul style="list-style-type: none"><li>• <a href="#">the section called “Broker default”</a></li></ul>
5 Mei 2021	<p>Menambahkan bagian berikut untuk menjelaskan kebijakan AWS terkelola untuk Amazon MQ dan pembaruan kebijakan ini:</p> <ul style="list-style-type: none"><li>• <a href="#">the section called “Kebijakan terkelola AWS”</a></li></ul>
16 Februari 2021	<p>Bagian tutorial tambahan berikut untuk Amazon MQ for RabbitMQ:</p> <ul style="list-style-type: none"><li>• <a href="#">the section called “Menyelesaikan sinkronisasi antrean yang dijeda”</a></li></ul>


Tanggal	Pembaruan Dokumentasi
4 November 2020	<ul style="list-style-type: none"><li>• Bagian berikut ditambahkan untuk mendokumentasikan dukungan Amazon MQ for RabbitMQ:<ul style="list-style-type: none"><li>• <a href="#">the section called “Membuat dan menghubungkan ke broker RabbitMQ”</a></li><li>• <a href="#">the section called “Tutorial RabbitMQ”</a></li><li>• <a href="#">the section called “Amazon MQ untuk praktik terbaik RabbitMQ”</a></li><li>• <a href="#">the section called “Mesin RabbitMQ”</a></li><li>• <a href="#">the section called “Mengonfigurasi Amazon MQ untuk RabbitMQ”</a></li><li>• <a href="#">the section called “Menggunakan peran yang terhubung dengan layanan”</a></li></ul></li><li>• Revisi tambahan untuk bab dan bagian panduan yang ada dibuat secara akurat untuk mendokumentasikan dukungan Amazon MQ for RabbitMQ.</li></ul>
16 Desember 2019	<ul style="list-style-type: none"><li>• Bagian berikut ditambahkan:<ul style="list-style-type: none"><li>• <a href="#">Storage</a></li><li>• <a href="#">Pilih jenis penyimpanan broker yang tepat untuk throughput terbaik</a></li></ul></li><li>• Merevisi informasi di bagian berikut:<ul style="list-style-type: none"><li>• <a href="#">Pialang</a></li><li>• <a href="#">Broker instance types</a></li><li>• <a href="#">Broker instans tunggal Amazon MQ</a></li><li>• <a href="#">Broker aktif/siaga Amazon MQ untuk ketersediaan tinggi</a></li><li>• <a href="#">Create an ActiveMQ broker</a></li><li>• <a href="#">Creating and configuring a broker</a></li></ul></li></ul>
19 Juli 2019	<p>Konten yang dimodifikasi dan ditambahkan pada manajemen enkripsi di bagian berikut:</p> <ul style="list-style-type: none"><li>• <a href="#">Perlindungan data di Amazon MQ</a></li><li>• <a href="#">Enkripsi saat istirahat</a></li><li>• <a href="#">Enkripsi dalam transit</a></li><li>• <a href="#">EncryptionOptions</a></li></ul>

Tanggal	Pembaruan Dokumentasi
22 April 2019	<p>Bagian berikut ditambahkan untuk kebijakan berbasis tanda dan izin tingkat sumber daya:</p> <ul style="list-style-type: none"><li>• <a href="#">Cara kerja Amazon MQ dengan IAM</a></li><li>• <a href="#">Izin tingkat sumber daya untuk tindakan API Amazon MQ</a></li></ul>
4 Maret 2019	<p>Peningkatan dokumentasi untuk mengonfigurasi failover dinamis dan menyeimbangkan ulang klien untuk jaringan broker. Aktifkan failover dinamis dengan mengonfigurasi <code>transportConnectors</code> bersama dengan opsi konfigurasi <code>networkConnectors</code> .</p> <ul style="list-style-type: none"><li>• <a href="#">Failover Dinamis dengan Konektor Transportasi</a></li><li>• <a href="#">Jaringan broker Amazon MQ</a></li><li>• <a href="#">Amazon MQ Broker Configuration Parameters</a></li></ul>
5 Januari 2019	<p>Peningkatan dokumentasi di beberapa metrik per menit. Untuk informasi lebih lanjut, lihat hal berikut: <a href="#">Metrik tujuan ActiveMQ (antrean dan topik)</a>.</p>
19 Desember 2018	<ul style="list-style-type: none"><li>• Bagian berikut ditambahkan:<ul style="list-style-type: none"><li>• <a href="#">Jaringan broker Amazon MQ</a></li><li>• <a href="#">Creating and Configuring a Network of Brokers</a></li><li>• <a href="#">Mengonfigurasi Jaringan Broker dengan Benar</a></li><li>• <a href="#">networkConnector</a></li><li>• <a href="#">networkConnectionStartasinkron</a></li></ul></li><li>• Penambahan elemen pengumpulan anak <code>networkConnectors</code> ke bagian <a href="#">Elemen, Elemen Pengumpulan Anak, dan Elemen Anaknya yang Diizinkan dalam Konfigurasi Amazon MQ</a>.</li></ul>
11 Desember 2018	<p>Pembaruan dokumentasi untuk mencerminkan ketersediaan ActiveMQ versi 5.15.8.</p>
5 Desember 2018	<p>Penambahan bagian <a href="#">Penandaan sumber daya</a>.</p>




Tanggal	Pembaruan Dokumentasi
26 Oktober 2018	Penambahan bagian <a href="#">Hindari mulai ulang lambat dengan memulihkan transaksi XA yang disiapkan</a> .
15 Oktober 2018	Pembaruan bagian <a href="#">Quotas in Amazon MQ</a> .
1 Oktober 2018	Perbaiki informasi di bagian <a href="#">Langkah selanjutnya</a> .
27 September 2018	<ul style="list-style-type: none"> <li>• Penambahan bagian <a href="#">Mengedit versi mesin broker, jenis instans, CloudWatchlog, dan preferensi pemeliharaan</a>.</li> <li>• Bagian berikut diperbarui: <ul style="list-style-type: none"> <li>• <a href="#">Create an ActiveMQ broker</a></li> <li>• <a href="#">Configure Basic Broker Settings</a></li> </ul> </li> </ul>
18 September 2018	Penambahan catatan berikut ke bagian <a href="#">Membuat dan mengelola pengguna broker ActiveMQ</a> : 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.
31 Agustus 2018	<ul style="list-style-type: none"> <li>• Klarifikasi terminologi untuk broker aktif/siaga. Untuk informasi selengkapnya, lihat <a href="#">Broker aktif/siaga Amazon MQ untuk ketersediaan tinggi</a>.</li> <li>• Penyederhanaan terminologi untuk jendela pemeliharaan. Untuk informasi selengkapnya, lihat <a href="#">Siklus hidup konfigurasi broker Amazon MQ</a>.</li> <li>• Penulisan ulang bagian <a href="#">Configure Additional Broker Settings</a>.</li> <li>• Pembaruan bagian <a href="#">Metrik Amazon MQ untuk ActiveMQ</a> dan <a href="#">Listing brokers and viewing broker details</a>.</li> </ul>
15 Agustus 2018	Perbaiki informasi di bagian <a href="#">Create an ActiveMQ broker</a> .
13 Agustus 2018	Penambahan bagian <a href="#">Mengakses konsol web broker tanpa aksesibilitas publik</a> .

Tanggal	Pembaruan Dokumentasi
2 Agustus 2018	<ul style="list-style-type: none"> <li>• Penambahan bagian <a href="#">Memecahkan Masalah Konfigurasi CloudWatch Logs</a>.</li> <li>• Penambahan peringatan berikut di seluruh panduan ini: <div data-bbox="431 359 1508 674" style="border: 1px solid #f08080; border-radius: 10px; padding: 10px; margin: 10px 0;"> <p><b>⚠ Important</b></p> <p>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.</p> </div> </li> </ul>
1 Agustus 2018	<p>Koreksi informasi di bagian berikut:</p> <ul style="list-style-type: none"> <li>• <a href="#">Memahami struktur pencatatan di CloudWatch Logs</a></li> <li>• <a href="#">Connect a Java application to your broker</a></li> </ul>
31 Juli 2018	<ul style="list-style-type: none"> <li>• Pemindahan <a href="#">video demo 3 menit</a> ke bagian <a href="#">Getting Started with Amazon MQ</a>.</li> <li>• Penambahan <a href="#">video memulai 3 menit</a> ke bagian <a href="#">What is Amazon MQ?</a>.</li> </ul>
30 Juli 2018	<ul style="list-style-type: none"> <li>• Penambahan bagian <a href="#">Configuring Amazon MQ to publish logs to Amazon CloudWatch Logs</a>.</li> <li>• Pembaruan bagian <a href="#">Configure Additional Broker Settings</a>.</li> </ul>
19 Juli 2018	<ul style="list-style-type: none"> <li>• Penambahan bagian <a href="#">Logging Amazon MQ API calls using CloudTrail</a>.</li> </ul>
5 Juli 2018	<ul style="list-style-type: none"> <li>• Penambahan referensi silang elemen anak <code>authorizationEntry</code> ke bagian <a href="#">Selalu konfigurasi peta otorisasi</a>.</li> <li>• Klarifikasi informasi di bagian <a href="#">Mengintegrasikan broker ActiveMQ dengan LDAP</a>.</li> <li>• Klarifikasi informasi di bagian <a href="#">Throttling API</a>.</li> </ul>
29 Juni 2018	<ul style="list-style-type: none"> <li>• Pembaruan informasi di bagian <a href="#">Broker instance types</a>.</li> <li>• Penambahan bagian <a href="#">Memilih Tipe Instans Broker yang Tepat untuk Throughput Terbaik</a>.</li> </ul>

Tanggal	Pembaruan Dokumentasi
4 Juni 2018	<p>Selain GitHub, HTML, PDF, dan Kindle, catatan rilis Amazon MQ Developer Guide tersedia sebagai umpan RSS.</p> 
29 Mei 2018	<p>Perubahan berikut dibuat di bagian <a href="#">Working Java Example</a>:</p> <ul style="list-style-type: none"><li>• Penambahan contoh STOMP+WSS Java. Kode Java contoh STOMP+WSS terhubung ke broker, membuat antrian, serta memublikasikan dan menerima pesan.</li><li>• Peningkatan contoh MQTT Java.</li><li>• Memperbaiki contoh OpenWire Java.</li></ul>
24 Mei 2018	<p>Koreksi port titik akhir protokol tingkat kawat pada contoh MQTT Java di bagian <a href="#">Working Java Example</a>.</p>
22 Mei 2018	<p>Koreksi informasi di semua bagian dependensi Java.</p>
17 Mei 2018	<p>Perbaiki informasi di bagian <a href="#">Pengguna</a>.</p>
15 Mei 2018	<p>Perbaiki informasi di bagian <a href="#">Memastikan performa Amazon MQ yang efektif</a>.</p>
8 Mei 2018	<ul style="list-style-type: none"><li>• Ditempatkan dalam <a href="#">Referensi izin REST API Amazon MQ</a> di bagiannya sendiri.</li><li>• Pembuatan bagian <a href="#">Izin IAM yang Diperlukan untuk Membuat Broker Amazon MQ</a> dengan contoh kebijakan IAM kustom.</li></ul>
7 Mei 2018	<ul style="list-style-type: none"><li>• Klarifikasi di seluruh panduan ini bahwa jendela pemeliharaan broker berdurasi 2 jam. Untuk informasi selengkapnya, lihat <a href="#">Siklus hidup konfigurasi broker Amazon MQ</a>.</li><li>• Penambahan penjelasan mengapa izin <code>ec2:CreateNetworkInterface</code> dan <code>ec2:CreateNetworkInterfacePermission</code> diperlukan untuk membuat broker. Untuk informasi selengkapnya, lihat <a href="#">Autentikasi dan otorisasi API untuk Amazon MQ</a>.</li></ul>

Tanggal	Pembaruan Dokumentasi
1 Mei 2018	<p>Klarifikasi informasi tentang jendela pemeliharaan untuk broker aktif/siaga di bagian berikut:</p> <ul style="list-style-type: none"><li>• <a href="#">Broker aktif/siaga Amazon MQ untuk ketersediaan tinggi</a></li><li>• <a href="#">Creating and configuring a broker</a></li><li>• <a href="#">Creating and applying broker configurations</a></li></ul>
27 April 2018	<p>Penulisan ulang bagian berikut dan optimisasi kode Java contoh agar sesuai dengan rekomendasi untuk menggunakan pooling koneksi hanya bagi produsen, bukan konsumen:</p> <ul style="list-style-type: none"><li>• <a href="#">Selalu Gunakan Pooling Koneksi</a></li><li>• <a href="#">Membuat produsen pesan dan mengirimkan pesan</a></li><li>• <a href="#">Membuat konsumen pesan dan menerima pesan</a></li><li>• <a href="#">AmazonMQExample.java</a></li></ul>
26 April 2018	<p>Penambahan contoh MQTT Java ke bagian <a href="#">Working Java Example</a>. Kode Java contoh MQTT terhubung ke broker, membuat topik, serta memublikasikan dan menerima pesan.</p>
4 April 2018	<p>Penggantian nama bagian Berkomunikasi dengan Amazon MQ menjadi <a href="#">Terhubung ke Amazon MQ</a>.</p>
3 April 2018	<p>Klarifikasi dan koreksi informasi di bagian <a href="#">Menonaktifkan Penyimpanan dan Pengiriman Bersamaan untuk Antrean dengan Konsumen Lambat</a>.</p>
2 April 2018	<p>Pemindahan Penyimpanan dan Pengiriman Bersamaan untuk Antrean di bagian Amazon MQ ke bagian <a href="#">Menonaktifkan Penyimpanan dan Pengiriman Bersamaan untuk Antrean dengan Konsumen Lambat</a>.</p>

Tanggal	Pembaruan Dokumentasi
27 Maret 2018	<ul style="list-style-type: none"> <li>• Penggantian <a href="#">video peluncuran re:Invent</a> dengan <a href="#">video demo 3 menit</a> di bagian <a href="#">What is Amazon MQ?</a>.</li> <li>• Restrukturisasi bagian berikut: <ul style="list-style-type: none"> <li>• <a href="#">Broker Architecture</a></li> <li>• <a href="#">Cara Kerja Amazon MQ</a>.</li> </ul> </li> <li>• Pemindahan <a href="#">Siklus hidup konfigurasi broker Amazon MQ</a> dalam bagian <a href="#">Broker Architecture</a>.</li> </ul>
22 Maret 2018	<p>Klarifikasi pernyataan berikut melalui panduan ini: Amazon MQ mengenkripsi pesan saat istirahat dan dalam transit menggunakan kunci enkripsi yang dikelola dan disimpan dengan aman. Untuk informasi selengkapnya, lihat <a href="#">Panduan Developer AWS Encryption SDK</a>.</p>
19 Maret 2018	<p>Klarifikasi pernyataan berikut di seluruh panduan ini: 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.</p>
15 Maret 2018	<ul style="list-style-type: none"> <li>• Restrukturisasi bagian <a href="#">Amazon MQ Basic elements</a>.</li> </ul>
12 Maret 2018	<ul style="list-style-type: none"> <li>• Klarifikasi dan koreksi informasi di bagian <a href="#">Praktik terbaik keamanan untuk Amazon MQ</a> dan <a href="#">Terhubung ke Amazon MQ</a>.</li> <li>• Penambahan bagian <a href="#">Menonaktifkan Penyimpanan dan Pengiriman Bersamaan untuk Antrean dengan Konsumen Lambat</a>.</li> <li>• Pengelompokkan peringatan ke dalam kata pengantar untuk bagian <a href="#">Mengonfigurasi pengaturan broker lanjutan</a>.</li> </ul>
9 Maret 2018	<ul style="list-style-type: none"> <li>• Klarifikasi dan koreksi informasi di bagian <a href="#">Selalu konfigurasi peta otorisasi</a>.</li> <li>• Penambahan bagian <a href="#">authorizationEntry</a> dan pembaruan bagian <a href="#">KahaDB</a>.</li> </ul>
8 Maret 2018	<ul style="list-style-type: none"> <li>• Penambahan bagian <a href="#">Selalu konfigurasi peta otorisasi</a>.</li> <li>• Penambahan catatan tentang sufiks broker ke bagian <a href="#">Monitoring Amazon MQ using CloudWatch</a>.</li> </ul>

Tanggal	Pembaruan Dokumentasi
6 Maret 2018	<p>Penambahan catatan berikut di seluruh panduan ini:</p> <div data-bbox="399 302 1507 667" style="border: 1px solid #add8e6; border-radius: 10px; padding: 10px; margin: 10px 0;"> <p><b>Note</b></p> <p>Penggunaan tipe instans <code>mq.t2.micro</code> tunduk pada <a href="#">kredit CPU dan performa dasar</a>—dengan kemampuan untuk burst di atas tingkat dasar (untuk informasi selengkapnya, lihat metrik <a href="#">CpuCredit Balance</a>). Jika aplikasi Anda membutuhkan performa tetap, pertimbangkan untuk menggunakan tipe instans <code>mq.m5.large</code>.</p> </div>
1 Maret 2018	<ul style="list-style-type: none"> <li>• Penambahan metrik <code>CpuCreditBalance</code> ke bagian <a href="#">Metrik Amazon MQ untuk ActiveMQ</a>.</li> <li>• Penambahan bagian <a href="#">Atribut Elemen Anak Amazon MQ</a>.</li> <li>• Penambahan tautan dari elemen di bagian <a href="#">the section called “Elemen yang Diizinkan”</a> ke atributnya dan ke elemen pengumpulan anak.</li> <li>• Membuat koreksi pada AWS Glosarium di. GitHub</li> </ul>
28 Februari 2018	Tampilan gambar yang dikoreksi di GitHub.
27 Februari 2018	<p>Selain HTML, PDF, dan Kindle, Panduan Pengembang Amazon MQ tersedia di. GitHub Untuk meninggalkan umpan balik, pilih GitHub ikon di sudut kanan atas.</p> <div data-bbox="399 1314 708 1423" style="border: 1px solid #ccc; padding: 5px; margin: 10px 0;">  </div>
26 Februari 2018	<ul style="list-style-type: none"> <li>• Pembuatan wilayah yang konsisten di semua contoh dan diagram.</li> <li>• Tautan yang dioptimalkan ke AWS konsol dan halaman web produk.</li> </ul>

Tanggal	Pembaruan Dokumentasi
22 Februari 2018	<p>Klarifikasi dan koreksi informasi di bagian berikut:</p> <ul style="list-style-type: none"><li>• <a href="#">Lebih memilih broker tanpa aksesibilitas publik</a></li><li>• <a href="#">Selalu Gunakan Transportasi Failover untuk Terhubung ke Beberapa Titik Akhir Broker</a></li><li>• <a href="#">Autentikasi dan otorisasi API untuk Amazon MQ</a></li><li>• <a href="#">Mengintegrasikan broker ActiveMQ dengan LDAP</a></li></ul>
21 Februari 2018	<p>Koreksi kode Java di bagian berikut:</p> <ul style="list-style-type: none"><li>• <a href="#">Working Java Example</a></li><li>• <a href="#">Connect a Java application to your broker</a></li><li>• <a href="#">Selalu Gunakan Pooling Koneksi</a></li></ul>
20 Februari 2018	<p>Mengklarifikasi dan mengoreksi informasi di bagian <a href="#">Keamanan di Amazon MQ</a> dan Praktik Terbaik.</p>
19 Februari 2018	<ul style="list-style-type: none"><li>• Koreksi kode Java di bagian <a href="#">Selalu Gunakan Pooling Koneksi</a>.</li><li>• Merestrukturisasi dan memperluas bagian dan <a href="#">Keamanan di Amazon MQ</a> bagian Praktik Terbaik.</li></ul>
16 Februari 2018	<ul style="list-style-type: none"><li>• Penambahan bagian <a href="#">Praktik terbaik keamanan untuk Amazon MQ</a>.</li><li>• Pembaruan bagian <a href="#">Terhubung ke Amazon MQ</a>.</li><li>• Koreksi kode Java di bagian berikut:<ul style="list-style-type: none"><li>• <a href="#">Getting Started with Amazon MQ</a></li><li>• <a href="#">AmazonMQExample.java</a></li></ul></li></ul>
15 Februari 2018	<ul style="list-style-type: none"><li>• Merestrukturisasi dan memperluas bagian bagian Praktik Terbaik.</li><li>• Pembaruan bagian berikut:<ul style="list-style-type: none"><li>• <a href="#">Bagaimana cara memulai dengan Amazon MQ?</a></li><li>• <a href="#">Langkah selanjutnya</a> (Memulai)</li><li>• <a href="#">Related resources</a></li></ul></li></ul>

Tanggal	Pembaruan Dokumentasi
14 Februari 2018	<p>Pembaruan bagian berikut:</p> <ul style="list-style-type: none"><li>• <a href="#">Quotas in Amazon MQ</a></li><li>• <a href="#">Throttling API</a></li><li>• <a href="#">Keamanan di Amazon MQ</a></li></ul>
13 Februari 2018	<ul style="list-style-type: none"><li>• Pembaruan bagian <a href="#">Related resources</a>.</li><li>• Pembaruan bagian <a href="#">Quotas in Amazon MQ</a>.</li><li>• Penambahan bagian <a href="#">Kami ingin mendengar pendapat Anda</a>.</li></ul>
25 Januari 2018	<ul style="list-style-type: none"><li>• Perbaiki kesalahan di subbagian <a href="#">Menambahkan dependensi Java</a> pada bagian <a href="#">Working Java Example</a>.</li><li>• Izin untuk ID operasi REST <a href="#">RebootBroker</a> tercantum dengan benar sebagai <code>mq:RebootBroker</code> pada konsol IAM.</li></ul>
24 Januari 2018	<ul style="list-style-type: none"><li>• Penambahan bagian <a href="#">Jangan Pernah Memodifikasi atau Menghapus Antarmuka Jaringan Elastis Amazon MQ</a>.</li><li>• Pembaruan semua diagram di seluruh panduan ini.</li><li>• Penambahan tautan ke <a href="#">Referensi REST API Amazon MQ</a> di seluruh panduan ini dan tautan ke REST API tertentu ke bagian <a href="#">Autentikasi dan otorisasi API untuk Amazon MQ</a>.</li></ul>
19 Januari 2018	<p>Pembaruan informasi di bagian <a href="#">Sumber daya Amazon MQ for ActiveMQ</a>.</p>
18 Januari 2018	<p>Klarifikasi dan koreksi informasi di bagian <a href="#">Quotas in Amazon MQ</a>.</p>
17 Januari 2018	<p>Pemulihan kembali <a href="#">rekomendasi untuk memilih tujuan virtual melalui langganan tahan lama</a>, dengan penjelasan yang lebih baik.</p>
11 Januari 2018	<ul style="list-style-type: none"><li>• Panduan Developer Amazon MQ tersedia dalam format <a href="#">Kindle</a>, selain HTML dan <a href="#">PDF</a>.</li><li>• Klarifikasi dan koreksi informasi di bagian <a href="#">Autentikasi dan otorisasi API untuk Amazon MQ</a> dan <a href="#">Langkah 2: buat pengguna dan dapatkan AWS kredensialnya</a>.</li></ul>



Tanggal	Pembaruan Dokumentasi
3 Januari 2018	Penambahan <code>DescribeConfigurationRevision</code> ke bagian <a href="#">Autentikasi dan otorisasi API untuk Amazon MQ</a> .
15 Desember 2017	Menghapus rekomendasi terhadap langganan tahan lama dari bagian Praktik Terbaik.
8 Desember 2017	<ul style="list-style-type: none"> <li>• Penambahan prasyarat <a href="#">Mengaktifkan koneksi masuk</a> ke bagian <a href="#">Connecting a Java application to your broker</a> dan <a href="#">Working Java Example</a>.</li> <li>• Penambahan catatan berikut ke seluruh panduan ini: Saat ini, Anda tidak dapat menghapus konfigurasi.</li> </ul>
7 Desember 2017	<ul style="list-style-type: none"> <li>• Peningkatan kode di <a href="#">AmazonMQExample.java</a>.</li> <li>• Penambahan bagian <a href="#">Autentikasi dan otorisasi API untuk Amazon MQ</a>.</li> </ul>
5 Desember 2017	<ul style="list-style-type: none"> <li>• Klarifikasi dan koreksi informasi di bagian <a href="#">Monitoring Amazon MQ using CloudWatch</a>: <ul style="list-style-type: none"> <li>• Peningkatan deskripsi metrik.</li> <li>• Penambahan subbagian <a href="#">Metrik Amazon MQ untuk ActiveMQ</a> dan <a href="#">Dimensi untuk metrik broker ActiveMQ</a>.</li> </ul> </li> <li>• Penambahan video "Memperkenalkan Amazon MQ" ke bagian <a href="#">What is Amazon MQ?</a>.</li> </ul>
4 Desember 2017	<ul style="list-style-type: none"> <li>• Klarifikasi informasi berikut di bagian <a href="#">Penyimpanan Data</a>: Kapasitas penyimpanan per broker adalah 200 GB.</li> <li>• Penambahan <a href="#">Prasyarat</a> ke bagian <a href="#">Working Java Example</a>. (Paket <code>activemq-client.jar</code> dan <code>activemq-pool.jar</code> diperlukan agar contoh dapat berfungsi. Untuk informasi selengkapnya, lihat <a href="#">Connecting a Java application to your broker</a>).</li> </ul>

Tanggal	Pembaruan Dokumentasi
1 Desember 2017	<ul style="list-style-type: none"><li>• Pembaruan dan peningkatan screenshot di semua tutorial.</li><li>• Klarifikasi penjelasan berikut di seluruh panduan ini: Pembuatan perubahan pada revisi konfigurasi atau pengguna ActiveMQ tidaksegera menerapkan perubahan. Untuk menerapkan perubahan Anda, Anda harus menunggu jendela pemeliharaan berikutnya atau <a href="#">reboot broker</a>. Untuk informasi selengkapnya, lihat <a href="#">Siklus hidup konfigurasi broker Amazon MQ</a>.</li></ul>

# AWSGlosarium

Untuk AWS terminologi terbaru, lihat [AWSglosarium di Referensi](#). Glosarium AWS

Terjemahan disediakan oleh mesin penerjemah. Jika konten terjemahan yang diberikan bertentangan dengan versi bahasa Inggris aslinya, utamakan versi bahasa Inggris.