aws

Entwicklerhandbuch

Amazon MQ

Amazon MQ: Entwicklerhandbuch

Table of Contents

Was ist Amazon MQ?

Amazon MQ ist ein verwalteter Message Broker-Service, der die Migration zu einem Message Broker in der Cloud erleichtert. Mit einem Message Broker können Software-Anwendungen und -Komponenten mithilfe verschiedener Programmiersprachen, Betriebssysteme und formeller Messaging-Protokolle miteinander kommunizieren. Derzeit unterstützt Amazon MQ die Engine-Typen Apache ActiveMQ Classic und RabbitMQ.

Amazon MQ funktioniert mit Ihren vorhandenen Anwendungen und Services, ohne dass Sie Ihr eigenes Messaging-System verwalten, betreiben oder pflegen müssen.

Themen

Wie unterscheidet sich Amazon MQ von Amazon SQS oder Amazon? SNS

Amazon MQ ist ein verwalteter Message Broker-Service, der Kompatibilität mit vielen beliebten Message Brokern bietet. Wir empfehlen Amazon MQ für die Migration von Anwendungen von bestehenden Message Brokern, die auf Kompatibilität mit Protokollen APIs wie JMS AMQP 0-9-1, AMQP 1.0,, und angewiesen sind. MQTT OpenWire STOMP

Amazon SQS und Amazon SNS sind Warteschlangen- und Themendienste, die hochgradig skalierbar und einfach zu bedienen sind und für die Sie keine Nachrichtenbroker einrichten müssen. Wir empfehlen diese Dienste für neue Anwendungen, die von nahezu unbegrenzter Skalierbarkeit profitieren und einfach sindAPIs.

Wie sehen meine ersten Schritte mit Amazon MQ aus?

- Informationen zum Erstellen Ihres ersten Brokers mit Amazon MQ finden Sie unter oder Erste Schritte: Einen ActiveMQ-Broker erstellen und eine Verbindung zu ihm herstellen Erste Schritte: Einen RabbitMQ-Broker erstellen und eine Verbindung zu ihm herstellen

- Weitere Informationen über Richtlinien und Einschränkungen, die Ihnen dabei helfen, den größten Nutzen aus Amazon MQ zu ziehen, finden Sie unter Working with Amazon MQ for ActiveMQ und Working with Amazon MQ for RabbitMQ.
- Weitere Informationen zu Amazon MQ REST APIs finden Sie in der Amazon MQ MQ-Referenz REST API.
- Weitere Informationen zu Amazon AWS CLI MQ-Befehlen finden Sie unter Amazon MQ in der AWS CLI Befehlsreferenz.

Bitte geben Sie uns Feedback

Wir freuen uns über Ihr Feedback. Um Kontakt mit uns aufzunehmen, besuchen Sie das Amazon MQ-Diskussionsforum.

Einrichten von Amazon MQ

Bevor Sie Amazon MQ verwenden können, müssen Sie die folgenden Schritte ausführen.

Themen

Schritt 1: Voraussetzungen

Melde dich an für ein AWS-Konto

Wenn Sie noch keine haben AWS-Konto, führen Sie die folgenden Schritte aus, um eine zu erstellen.

Um sich für eine anzumelden AWS-Konto

1. Öffnen Sie https://portal.aws.amazon.com/billing/die Anmeldung.
2. Folgen Sie den Online-Anweisungen.

   Bei der Anmeldung müssen Sie auch einen Telefonanruf entgegennehmen und einen Verifizierungscode über die Telefontasten eingeben.

   Wenn Sie sich für eine anmelden AWS-Konto, Root-Benutzer des AWS-Kontoswird eine erstellt. Der Root-Benutzer hat Zugriff auf alle AWS -Services und Ressourcen des Kontos. Als bewährte Sicherheitsmethode weisen Sie einem Administratorbenutzer Administratorzugriff zu und verwenden Sie nur den Root-Benutzer, um Aufgaben auszuführen, die Root-Benutzerzugriff erfordern.

AWS sendet Ihnen nach Abschluss des Anmeldevorgangs eine Bestätigungs-E-Mail. Du kannst jederzeit deine aktuellen Kontoaktivitäten einsehen und dein Konto verwalten, indem du zu https://aws.amazon.com/gehst und Mein Konto auswählst.

Erstellen eines Benutzers mit Administratorzugriff

Nachdem Sie sich für einen angemeldet haben AWS-Konto, sichern Sie Ihren Root-Benutzer des AWS-Kontos AWS IAM Identity Center, aktivieren und erstellen Sie einen Administratorbenutzer, sodass Sie den Root-Benutzer nicht für alltägliche Aufgaben verwenden.

Sichern Sie Ihre Root-Benutzer des AWS-Kontos

1. Melden Sie sich AWS Management Consoleals Kontoinhaber an, indem Sie Root-Benutzer auswählen und Ihre AWS-Konto E-Mail-Adresse eingeben. Geben Sie auf der nächsten Seite Ihr Passwort ein.

   Hilfe bei der Anmeldung mit dem Root-Benutzer finden Sie unter Anmelden als Root-Benutzer im AWS-Anmeldung Benutzerhandbuch zu.
2. Aktivieren Sie die Multi-Faktor-Authentifizierung (MFA) für Ihren Root-Benutzer.

   Anweisungen finden Sie im Benutzerhandbuch unter Aktivieren eines virtuellen MFA Geräts für Ihren AWS-Konto IAM Root-Benutzer (Konsole).

Erstellen eines Benutzers mit Administratorzugriff

1. Aktivieren Sie IAM Identity Center.

   Anweisungen finden Sie unter Aktivieren AWS IAM Identity Center im AWS IAM Identity Center Benutzerhandbuch.
2. Gewähren Sie einem Benutzer in IAM Identity Center Administratorzugriff.

   Ein Tutorial zur Verwendung von IAM-Identity-Center-Verzeichnis als Identitätsquelle finden Sie unter Benutzerzugriff mit der Standardeinstellung konfigurieren IAM-Identity-Center-Verzeichnis im AWS IAM Identity Center Benutzerhandbuch.

Anmelden als Administratorbenutzer

- Um sich mit Ihrem IAM Identity Center-Benutzer anzumelden, verwenden Sie die Anmeldung, URL die an Ihre E-Mail-Adresse gesendet wurde, als Sie den IAM Identity Center-Benutzer erstellt haben.

  Hilfe bei der Anmeldung mit einem IAM Identity Center-Benutzer finden Sie im AWS-Anmeldung Benutzerhandbuch unter Anmeldung beim AWS Zugriffsportal.

Weiteren Benutzern Zugriff zuweisen

1. Erstellen Sie in IAM Identity Center einen Berechtigungssatz, der der bewährten Methode zur Anwendung von Berechtigungen mit den geringsten Rechten folgt.

   Anweisungen hierzu finden Sie unter Berechtigungssatz erstellen im AWS IAM Identity Center Benutzerhandbuch.

2. Weisen Sie Benutzer einer Gruppe zu und weisen Sie der Gruppe dann Single Sign-On-Zugriff zu.

   Eine genaue Anleitung finden Sie unter Gruppen hinzufügen im AWS IAM Identity Center Benutzerhandbuch.

Schritt 2: Erstellen Sie einen Benutzer und holen Sie sich Ihre Anmeldeinformationen AWS

Benutzer benötigen programmatischen Zugriff, wenn sie mit AWS außerhalb des AWS Management Console interagieren möchten. Die Art und Weise, wie programmatischer Zugriff gewährt wird, hängt von der Art des Benutzers ab, der zugreift. AWS

Um Benutzern programmgesteuerten Zugriff zu gewähren, wählen Sie eine der folgenden Optionen.

Welcher Benutzer benötigt programmgesteuerten Zugriff?	Bis	Von
Mitarbeiteridentität (In IAM Identity Center verwaltete Benutzer)	Verwenden Sie temporäre Anmeldeinformationen, um programmatische Anfragen an das AWS CLI AWS SDKs, oder AWS APIs zu signieren.	Befolgen Sie die Anweisungen für die Schnittstelle, die Sie verwenden möchten. • Informationen zu den AWS CLI finden Sie unter Konfiguration der AWS CLI zur Verwendung AWS IAM Identity Center im AWS Command Line Interface Benutzerhandbuch.

Welcher Benutzer benötigt programmgesteuerten Zugriff?	Bis	Von
		• Informationen zu AWS SDKs Tools und AWS APIs finden Sie unter IAMIdentity Center-Authentifizierung im Referenzhandbuch AWS SDKsund im Tools-Referenzhandbuch.
IAM	Verwenden Sie temporäre Anmeldeinformationen, um programmatische Anfragen an das AWS CLI AWS SDKs, oder AWS APIs zu signieren.	Folgen Sie den Anweisungen unter Verwenden temporärer Anmeldeinformationen mit AWS Ressourcen im IAMBenutzerhandbuch.

Welcher Benutzer benötigt programmgesteuerten Zugriff?	Bis	Von
IAM	(Nicht empfohlen) Verwenden Sie langfristige Anmeldeinformationen, um programmatische Anfragen an das AWS CLI AWS SDKs, oder AWS APIs zu signieren.	Befolgen Sie die Anweisungen für die Schnittstelle, die Sie verwenden möchten. • Informationen dazu AWS CLI finden Sie unter Authentifizierung mithilfe von IAM Benutzeranmeldedaten im AWS Command Line Interface Benutzerhandbuch. • Informationen zu AWS SDKs und Tools finden Sie unter Authentifizieren mit langfristigen Anmeldeinformationen im Referenzhandbuch AWS SDKsund im Tools-Referenzhandbuch. • Weitere Informationen finden Sie unter Verwaltung von Zugriffsschlüsseln für IAM Benutzer im IAMBenutzerhandbuch. AWS APIs

Schritt 3: Vorbereiten der Verwendung des Beispiel-Codes

Die folgenden Tutorials zeigen, wie Sie mithilfe von Amazon MQ-Brokern arbeiten können und AWS Management Console wie Sie programmatisch eine Verbindung zu Ihren Amazon MQ for ActiveMQ- und Amazon MQ for RabbitMQ-Brokern herstellen. Wenn Sie den Beispiel-Code verwenden möchten, müssen Sie das Java Standard Edition Development Kit installieren und einige Änderungen am Code vornehmen.

Sie können Broker auch programmgesteuert mit Amazon RESTAPIMQ und verwalten. AWS SDKs

Nächste Schritte

Sie sind nun bereit für die ersten Schritte mit Amazon MQ und können einen Broker erstellen. Abhängig von Ihrem Broker-Engine-Typ können Sie dann eine Java-Anwendung mit Ihrem Amazon MQ for ActiveMQ-Broker verbinden oder die RabbitMQ-Java-Clientbibliothek verwenden, um eine JVM basierte Anwendung mit Ihrem Amazon MQ for RabbitMQ-Broker zu verbinden.

Erste Schritte: Einen ActiveMQ-Broker erstellen und eine Verbindung zu ihm herstellen

Ein Broker ist eine Message-Broker-Umgebung, die auf Amazon MQ ausgeführt wird. Dies ist der Grundblock für Amazon MQ. Die kombinierte Beschreibung der Broker-Instanceclass(`m5`,`t3`) undsize(`large`,`micro`) ist einBroker-Instance-Typ(zum Beispiel`mq.m5.large`). Weitere Informationen finden Sie unter Was ist ein Amazon MQ for ActiveMQ-Broker?.

Schritt 1: Erstellen Sie einen ActiveMQ-Broker

Die erste und häufigste Amazon-MQ-Aufgabe ist das Erstellen eines Brokers. Das folgende Beispiel zeigt, wie Sie den verwenden können AWS Management Console , um einen einfachen Broker zu erstellen.

1. Melden Sie sich bei der Amazon MQ-Konsole an.
2. Wählen Sie auf der Seite Broker-Engine auswählen die Option Apache ActiveMQ aus.
3. Auf der Seite Select deployment and storage (Auswählen von Bereitstellung und Speicher), tun sie das Folgende im Bereich Deployment mode and storage type (Bereitstellungsmodus und Speichertyp):

   a. Wählen Sie den Bereitstellungsmodus (z. B. Aktiv/Standby-Broker). Weitere Informationen finden Sie unter Bereitstellungsoptionen für Amazon MQ für ActiveMQ-Broker.

      - Ein Single-Instance-Broker besteht aus einem Broker in einer Availability Zone. Der Broker kommuniziert mit Ihrer Anwendung und mit einem Amazon- EBS oder EFS Amazon-Speichervolumen. Weitere Informationen finden Sie unter Option 1: Amazon MQ-Broker mit einer einzigen Instanz.
      - Ein Aktiv/Standby-Broker für hohe Verfügbarkeit besteht aus zwei Brokern in zwei verschiedenen Availability Zones, die in einem redundanten Paar konfiguriert sind. Diese Broker kommunizieren synchron mit Ihrer Anwendung und mit AmazonEFS. Weitere Informationen finden Sie unter Option 2: Amazon MQ Active/Standby-Broker für hohe Verfügbarkeit.
      - Weitere Informationen zu den Beispiel-Blueprints für ein Netzwerk von Brokern finden Sie unterBeispiel-Vorlagen.

   b. Wählen Sie den Speichertyp (z. B. EBS). Weitere Informationen finden Sie unter Storage.

> **Note**
>
> Amazon EBS repliziert Daten innerhalb einer einzigen Availability Zone und unterstützt den ActiveMQ-Aktiv-/Standby-Bereitstellungsmodus nicht.

c. Wählen Sie Weiter aus.

4. Gehen Sie auf der Seite Einstellungen konfigurieren im Abschnitt Details wie folgt vor:

   a. Geben Sie den Broker-Namen ein.

   > **Important**
   >
   > Fügen Sie den Namen der Makler keine personenbezogenen Daten (PII) oder andere vertrauliche oder sensible Informationen hinzu. Broker-Namen sind für andere AWS Dienste zugänglich, einschließlich CloudWatch Logs. Brokernamen sind nicht für private oder sensible Daten gedacht.

   b. Wählen Sie den Broker-Instance-Typ (z. B. mq.m5.large). Weitere Informationen finden Sie unter Broker instance types.

5. Geben Sie im Abschnitt Zugriff auf ActiveMQ-Webkonsole einen Benutzernamen und ein Passwort an. Die folgenden Einschränkungen gelten in Bezug auf Benutzernamen und Passwörter des Brokers:

   - Ihr Benutzername darf nur alphanumerische Zeichen, Bindestriche, Punkte, Unterstriche und Tilden (- . _ ~) enthalten.
   - Ihr Psswort muss mindestens 12 Zeichen lang sein, muss mindestens 4 eindeutige Zeichen enthalten und darf keine Kommas, Doppelpunkte oder Gleichheitszeichen (,:=) enthalten.

   > **Important**
   >
   > Fügen Sie den Broker-Benutzernamen keine personenbezogenen Daten (PII) oder andere vertrauliche oder sensible Informationen hinzu. Broker-Benutzernamen sind für andere AWS Dienste zugänglich, einschließlich CloudWatch Logs. Broker-Benutzernamen sind nicht für private oder sensible Daten gedacht.

6. Wählen Sie Deploy (Bereitstellen) aus.

Während Amazon MQ Ihren Broker erstellt, zeigt er denWird erstelltStatus an.

Die Erstellung eines Brokers dauert etwa 15 Minuten.

Wenn Ihr Broker erfolgreich erstellt wurde, zeigt Amazon MQ den Running-Status (Ausführung) an.

7. Wählen Sie aus.***MyBroker***.

   Auf der ***MyBroker***Notieren Sie sich auf der Seite im Bereich Connect die ActiveMQ-Webkonsole Ihres BrokersURL, zum Beispiel:

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

   Beachten Sie auch dieWire-Level-Protokoll-Endpunkte. Das Folgende ist ein Beispiel für einen OpenWire Endpunkt:

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

Schritt 2: Connect eine Java-Anwendung mit Ihrem Broker

Nachdem Sie einen Amazon MQ ActiveMQ Broker erstellt haben, können Sie Ihre Anwendung mit ihm verbinden. Die folgenden Beispiele zeigen, wie Sie den Java Message Service (JMS) verwenden können, um eine Verbindung zum Broker herzustellen, eine Warteschlange zu erstellen und eine Nachricht zu senden. Ein vollständiges, funktionierendes Java-Beispiel finden Sie unterWorking Java Example.

Sie können unter Verwendung verschiedener ActiveMQ-Clients eine Verbindung zu ActiveMQ-Brokern einrichten. Wir empfehlen die Verwendung des ActiveMQ-Clients.

Voraussetzungen

VPCAttribute aktivieren

> **Note**
>
> Sie können die öffentliche Zugänglichkeit für Ihre vorhandenen Amazon-MQ-Broker nicht deaktivieren.

Um sicherzustellen, dass Ihr Broker in Ihrem erreichbar istVPC, müssen Sie die `enableDnsSupport` VPC Attribute `enableDnsHostnames` und aktivieren. Weitere Informationen finden Sie unter DNSSupport VPC in Ihrem VPC Amazon-Benutzerhandbuch.

Eingehende Verbindungen aktivieren

Verwenden Sie als Nächstes die folgenden Anweisungen, um eingehende Verbindungen für Ihren Broker zu aktivieren.

1. Melden Sie sich bei der Amazon-MQ-Konsole an.
2. Wählen Sie aus der Brokerliste den Namen Ihres Brokers aus (z. B. MyBroker).
3. Auf der ***MyBroker***Notieren Sie sich im Abschnitt Verbindungen die Adressen und Ports der Webkonsole URL und der Wire-Level-Protokolle des Brokers.
4. Wählen Sie im Abschnitt Details unter Sicherheit und Netzwerk den Namen Ihrer Sicherheitsgruppe oder

   .

   Die Seite „Sicherheitsgruppen“ des EC2 Dashboards wird angezeigt.
5. Wählen Sie in der Liste der Sicherheitsgruppen Ihre Sicherheitsgruppe.
6. Klicken Sie unten auf der Seite auf Inbound (Eingehend) und anschließend auf Edit (Bearbeiten).
7. Fügen Sie im Dialogfeld „Regeln für eingehenden Datenverkehr bearbeiten“ eine Regel für jeden URL oder Endpunkt hinzu, auf den Sie öffentlich zugreifen möchten (das folgende Beispiel zeigt, wie Sie dies für eine Broker-Webkonsole tun können).

   a. Klicken Sie auf Add Rule (Regel hinzufügen).

   b. Wählen Sie für Typ die Option Benutzerdefiniert TCP aus.

   c. Für Port-Bereich, geben Sie den Port der Webkonsole ein (`8162`).

   d. Für Source (Quelle), lassen Sie Custom (Benutzerdefiniert) ausgewählt, und geben Sie dann die IP-Adresse des Systems ein, auf das auf die Webkonsole zugegriffen werden soll (z. B. `192.0.2.1`) enthalten.

   e. Wählen Sie Save.

   Ihr Broker kann nun eingehende Verbindungen akzeptieren.

Java-Abhängigkeiten hinzufügen

Fügen Sie dem Pfad für Ihre Java-Build-Klasse die Pakete `activemq-client.jar` und `activemq-pool.jar`hinzu. Das folgende Beispiel zeigt diese Abhängigkeiten in der `pom.xml`-Datei eines Maven-Projekts.

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

Weitere Informationen über `activemq-client.jar` finden Sie unter Ursprüngliche Konfiguration in der Apache ActiveMQ-Dokumentation.

> **Important**
>
> Im folgenden Beispielcode laufen Hersteller und Verbraucher in einem einzigen Thread. Stellen Sie für Produktionssysteme (oder zum Testen des Failovers von Broker-Instances) sicher, dass Ihre Produzenten und Verbraucher auf separaten Hosts oder Threads ausgeführt werden.

Erstellen eines Nachrichtenproduzenten und Senden einer Nachricht

Stellen Sie als Nächstes sicher, dass Ihr Broker eine Nachricht empfangen kann, indem Sie einen Nachrichtengenerator erstellen und eine Nachricht senden.

1. Erstellen Sie mithilfe des Endpunkts Ihres Brokers eine JMS gepoolte Verbindungs-Factory für den Nachrichtenproduzenten und rufen Sie dann die `createConnection` Methode für die Factory auf.

> **Note**
>
> Für einen Aktiv-/Standby-Broker bietet Amazon MQ zwei ActiveMQ-Web-KonsolenURLs, von denen jedoch jeweils nur eine aktiv URL ist. Ebenso stellt Amazon MQ zwei Endpunkte für jedes Wire-Level-Protokoll bereit, jedoch ist jeweils nur ein Endpunkt in jedem Paar aktiv. Die `-1`- und `-2`-Suffixe bezeichnen ein redundantes Paar. Weitere Informationen finden Sie unter Bereitstellungsoptionen für Amazon MQ für ActiveMQ-Broker).
>
> Für Drahtebene Protokollendpunkte können Sie zulassen, dass Ihre Anwendung eine Verbindung zu einem beliebigen Endpunkt herstellen kann, indem Sie dieFailover-Transport verwenden.

```
// 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**
>
> Nachrichtenproduzenten sollten immer die`PooledConnectionFactory`-Klasse. Weitere Informationen finden Sie unter Verwenden Sie immer Verbindungspools.

2. Erstellen Sie eine Sitzung, eine Warteschlange namens`MyQueue`und einen Nachrichtenproduzenten.

```
// 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. Erstellen der Nachrichtenzeichenfolge`"Hello from Amazon MQ!"`Dann senden Sie die Nachricht.

```
// 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. Bereinigen Sie den Produzenten.

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

Erstellen eines Nachrichtenkonsumenten und Empfangen der Nachricht

Nachdem Sie einen Producer erstellt haben, erstellen Sie einen Consumer, um zu überprüfen, ob er die Nachricht empfangen kann.

1. Erstellen Sie mithilfe des Endpunkts Ihres Brokers eine JMS Verbindungs-Factory für den Nachrichtenproduzenten und rufen Sie dann die `createConnection` Methode für die Factory auf.

```
// 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**
>
> Die Nachrichtenkonsumenten sollten nie die `PooledConnectionFactory`-Klasse verwenden. Weitere Informationen finden Sie unter Verwenden Sie immer Verbindungspools.

2. Erstellen Sie eine Sitzung, eine Warteschlange namens`MyQueue`und einem Nachrichtenverbraucher.

```
// 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. Beginnen Sie, auf Nachrichten zu warten und die Nachricht zu erhalten, wenn sie eintrifft.

```
// 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**
>
> Im Gegensatz zu AWS Messaging-Diensten (wie AmazonSQS) ist der Verbraucher ständig mit dem Broker verbunden.

4. Schließen Sie den Verbraucher, die Sitzung und die Verbindung.

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

Schritt 3: (Optional) Connect zu einer AWS Lambda Funktion herstellen

AWS Lambda kann eine Verbindung zu Ihrem Amazon MQ-Broker herstellen und Nachrichten von diesem empfangen. Wenn Sie einen Broker mit Lambda verbinden, erstellen Sie eine Ereignisquellen-Zuweisung, der Nachrichten aus einer Warteschlange liest und die Funktionsynchron. Die von Ihnen erstellte Ereignisquellenzuordnung liest Nachrichten von Ihrem Broker stapelweise und konvertiert sie in eine Lambda-Payload in Form eines Objekts. JSON

So verbinden Sie Ihren Broker mit einer Lambda Funktion

1. Fügen Sie Ihrer IAM Lambda-Funktionsausführungsrolle die folgenden Rollenberechtigungen hinzu.

   - mq: DescribeBroker
   - ec2: CreateNetworkInterface
   - ec2: DeleteNetworkInterface
   - ec2: DescribeNetworkInterfaces
   - ec2: DescribeSecurityGroups
   - ec2: DescribeSubnets
   - ec2: DescribeVpcs
   - Protokolle: CreateLogGroup
   - Protokolle: CreateLogStream

- Protokolle: PutLogEvents
- Verwalter von Geheimnissen: GetSecretValue

> **Note**
>
> Ohne die erforderlichen IAM Berechtigungen kann Ihre Funktion keine Datensätze aus Amazon MQ MQ-Ressourcen erfolgreich lesen.

2. (Optional) Wenn Sie einen Broker ohne öffentliche Zugänglichkeit erstellt haben, müssen Sie einen der folgenden Schritte ausführen, damit Lambda eine Verbindung zu Ihrem Broker herstellen kann:

   - Konfigurieren Sie ein NAT Gateway pro öffentlichem Subnetz. Weitere Informationen finden Sie unter Internet- und VPC Dienstzugriff für verbundene Funktionen im AWS Lambda Entwicklerhandbuch.
   - Stellen Sie mithilfe eines VPC Endpunkts eine Verbindung zwischen Ihrer Amazon Virtual Private Cloud (AmazonVPC) und Lambda her. Ihr Amazon VPC muss auch eine Verbindung zu AWS Security Token Service (AWS STS) und Secrets Manager Manager-Endpunkten herstellen. Weitere Informationen finden Sie unter Configuring Interface VPC Endpoints for Lambda im AWS Lambda Developer Guide.

3. Konfigurieren Sie Ihren Broker als EreignisquelleVerwendung für eine Lambda -Funktion unter Verwendung der AWS Management Console. Sie können den Befehl auch verwenden. `create-event-source-mapping` AWS Command Line Interface

4. Schreiben Sie Code für Ihre Lambda Funktion, um die von Ihrem Broker verbrauchten Nachrichten zu verarbeiten. Die Lambda-Payload, die von der Ereignisquellen-Zuweisung abgerufen wird, hängt vom Modultyp des Brokers ab. Im Folgenden finden Sie ein Beispiel für eine Lambda-Nutzlast für eine Warteschlange in Amazon MQ für RabbitMQ.

> **Note**
>
> Im Beispiel ist `testQueue` der Name der Warteschlange.

```
{
  "eventSource": "aws:amq",
```

```
  "eventSourceArn": "arn:aws:mq:us-
west-2:112556298976:broker:test:b-9bcfa592-423a-4942-879d-eb284b418fc8",
  "messages": {
    [
      {
        "messageID": "ID:b-9bcfa592-423a-4942-879d-eb284b418fc8-1.mq.us-
west-2.amazonaws.com-37557-1234520418293-4:1:1:1:1",
        "messageType": "jms/text-message",
        "data": "QUJDOkFBQUE=",
        "connectionId": "myJMSCoID",
        "redelivered": false,
        "destination": {
          "physicalname": "testQueue"
        },
        "timestamp": 1598827811958,
        "brokerInTime": 1598827811958,
        "brokerOutTime": 1598827811959
      },
      {
        "messageID": "ID:b-9bcfa592-423a-4942-879d-eb284b418fc8-1.mq.us-
west-2.amazonaws.com-37557-1234520418293-4:1:1:1:1",
        "messageType":"jms/bytes-message",
        "data": "3DTOOW7crj51prgVLQaGQ82S48k=",
        "connectionId": "myJMSCoID1",
        "persistent": false,
        "destination": {
          "physicalname": "testQueue"
        },
        "timestamp": 1598827811958,
        "brokerInTime": 1598827811958,
        "brokerOutTime": 1598827811959
      }
    ]
  }
}
```

Weitere Informationen zum Verbinden von Amazon MQ mit Lambda, zu den Optionen, die Lambda für eine Amazon-MQ-Ereignisquelle unterstützt, und zu Fehlern bei der Ereignisquellen-Zuweisung finden Sie unter Verwenden von Lambda mit Amazon MQ im AWS Lambda -Entwicklerhandbuch.

Erste Schritte: Einen RabbitMQ-Broker erstellen und eine Verbindung zu ihm herstellen

Ein Broker ist eine Message-Broker-Umgebung, die auf Amazon MQ ausgeführt wird. Dies ist der Grundblock für Amazon MQ. Die kombinierte Beschreibung der Broker-Instanceclass(`m5`,`t3`) undsize(`large`,`micro`) ist einBroker-Instance-Typ(zum Beispiel`mq.m5.large`). Weitere Informationen finden Sie unter Was ist ein Amazon MQ for RabbitMQ Broker?

Schritt 1: Erstellen Sie einen RabbitMQ-Broker

Die erste und häufigste Amazon MQ-Aufgabe ist das Erstellen eines Brokers. Das folgende Beispiel zeigt, wie Sie den verwenden können, um einen einfachen AWS Management Console Broker zu erstellen.

1. Melden Sie sich bei der Amazon MQ-Konsole an.
2. Klicken Sie auf derWählen Sie Broker-EngineWählen Sie auf der SeiteRabbitMQKlicken Sie auf und danach auf Weiter.
3. Klicken Sie auf derAuswählen des BereitstellungsmodusWählen Sie auf der SeiteBereitstellungsmodusZum BeispielCluster-BereitstellungKlicken Sie auf und danach auf Weiter.

   - Ein Single-Instance-Broker besteht aus einem Broker in einer Availability Zone hinter einem Network Load Balancer ()NLB. Der Broker kommuniziert mit Ihrer Anwendung und mit einem EBS Amazon-Speichervolumen. Weitere Informationen finden Sie unter Option 1: Einzelinstanz-Broker Amazon MQ für RabbitMQ.
   - Eine Bereitstellung von RabbitMQ-Clustern für hohe Verfügbarkeit ist eine logische Gruppierung von drei RabbitMQ-Broker-Knoten hinter einem Network Load Balancer, wobei jeder Benutzer, Warteschlangen und ein verteilter Status über mehrere Availability Zones (AZ) verfügt. Weitere Informationen finden Sie unter Option 2: Amazon MQ für die RabbitMQ-Clusterbereitstellung.
4. Auf der Seite Einstellungen konfigurieren geben Sie im Abschnitt Details Folgendes ein:

   a. Geben Sie den Broker-Namen ein.

> **Important**
>
> Fügen Sie den Namen der Makler keine personenbezogenen Daten (PII) oder andere vertrauliche oder sensible Informationen hinzu. Broker-Namen sind für andere AWS Dienste zugänglich, einschließlich CloudWatch Logs. Brokernamen sind nicht für private oder sensible Daten gedacht.

b. Wählen Sie den Broker-Instance-Typ (z. B. mq.m5.large). Weitere Informationen finden Sie unter Broker instance types.

> **Note**
>
> Der Abschnitt Zusätzliche Einstellungen bietet Optionen zum Aktivieren von CloudWatch Protokollen und zum Konfigurieren des Netzwerkzugriffs für Ihren Broker. Wenn Sie einen privaten RabbitMQ-Broker ohne öffentlichen Zugriff erstellen, müssen Sie eine Virtual Private Cloud (VPC) auswählen und eine Sicherheitsgruppe für den Zugriff auf Ihren Broker konfigurieren.

5. Klicken Sie auf derKonfigurieren der EinstellungenKlicken Sie auf der SeiteZugriff auf RabbitMQ-Abschnitt eineBenutzername:undPasswort. Die folgenden Einschränkungen gelten in Bezug auf Broker-Anmeldeinformationen:

   - Er darf nur alphanumerische Zeichen, Bindestriche, Punkte, Unterstriche und Tilden (-. _) enthalten. Dieser Wert darf keine Tilde (~) Zeichen enthalten. Amazon MQ verbietet die Verwendung von `guest` als Benutzernamen.
   - Dieser Wert muss mindestens 12 Zeichen lang sein, muss mindestens 4 eindeutige Zeichen enthalten und darf keine Kommas, Doppelpunkte oder Gleichheitszeichen (,:=) enthalten.

   > **Important**
   >
   > Fügen Sie den Benutzernamen von Brokern keine personenbezogenen Daten (PII) oder andere vertrauliche oder sensible Informationen hinzu. Broker-Benutzernamen sind für andere AWS Dienste zugänglich, einschließlich CloudWatch Logs. Broker-Benutzernamen sind nicht für private oder sensible Daten gedacht.

6. Wählen Sie Weiter.

7. Auf der Seite Überprüfen und erstellen können Sie Ihre Auswahl überprüfen und sie nach Bedarf bearbeiten.
8. Wählen Sie Create broker (Broker erstellen).

   Während Amazon MQ Ihren Broker erstellt, zeigt er den Wird erstellt-Status an.

   Die Erstellung eines Brokers dauert etwa 15 Minuten.

   Wenn Ihr Broker erfolgreich erstellt wurde, zeigt Amazon MQ den Running-Status (Ausführung) an.
9. Wählen Sie aus.***MyBroker***.

   Auf der ***MyBroker***Notieren Sie sich auf der Seite im Bereich Connect die RabbitMQ-Webkonsole Ihres BrokersURL, zum Beispiel:

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

   Notieren Sie sich auch den sicheren Endpunkt Ihres Brokers. AMQP Es folgt ein Beispiel für ein`amqps`Endpunkt Zuweisen auf Listener-Port`5671`.

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

Schritt 2: Connect eine JVM basierte Anwendung mit Ihrem Broker

Nachdem Sie einen RabbitMQ-Broker erstellt haben, können Sie Ihre Anwendung mit ihm verbinden. Die folgenden Beispiele zeigen, wie Sie die RabbitMQ-Client-Bibliothek, um eine Verbindung zu Ihrem Broker zu erstellen, eine Warteschlange zu erstellen und eine Nachricht zu senden. Sie können sich mit RabbitMQ-Brokern verbinden, indem Sie unterstützte RabbitMQ-Client-Bibliotheken für eine Vielzahl von Sprachen verwenden. Weitere Informationen zu unterstützten RabbitMQ-Client-Bibliotheken finden Sie unterRabbitMQ-Client-Bibliotheken und Entwickler-Tools.

Voraussetzungen

> **Note**
>
> Die folgenden Schritte gelten nur für RabbitMQ-Broker, die ohne öffentliche Zugänglichkeit erstellt wurden. Wenn Sie einen Broker mit öffentlicher Barrierefreiheit erstellen, können Sie ihn überspringen.

Aktivieren Sie VPC Attribute

Um sicherzustellen, dass Ihr Broker in Ihrem erreichbar istVPC, müssen Sie die `enableDnsSupport` VPC Attribute `enableDnsHostnames` und aktivieren. Weitere Informationen finden Sie unter DNSSupport VPC in Ihrem VPC Amazon-Benutzerhandbuch.

Eingehende Verbindungen aktivieren

1. Melden Sie sich bei der Amazon MQ-Konsole an.
2. Wählen Sie aus der Brokerliste den Namen Ihres Brokers aus (z. B. MyBroker).
3. Auf der ***MyBroker***Notieren Sie sich im Abschnitt Verbindungen die Adressen und Ports der Webkonsole URL und der Wire-Level-Protokolle des Brokers.
4. Wählen Sie im Abschnitt Details unter Sicherheit und Netzwerk den Namen Ihrer Sicherheitsgruppe oder

   .

   Die Seite „Sicherheitsgruppen“ des EC2 Dashboards wird angezeigt.
5. Wählen Sie in der Liste der Sicherheitsgruppen Ihre Sicherheitsgruppe.
6. Klicken Sie unten auf der Seite auf Inbound (Eingehend) und anschließend auf Edit (Bearbeiten).
7. Fügen Sie im Dialogfeld „Regeln für eingehenden Datenverkehr bearbeiten“ eine Regel für jeden URL oder Endpunkt hinzu, auf den Sie öffentlich zugreifen möchten (das folgende Beispiel zeigt, wie Sie dies für eine Broker-Webkonsole tun können).

   a. Klicken Sie auf Add Rule (Regel hinzufügen).

   b. Wählen Sie für Typ die Option Benutzerdefiniert TCP aus.

   c. Für Source (Quelle), lassen Sie Custom (Benutzerdefiniert) ausgewählt, und geben Sie dann die IP-Adresse des Systems ein, auf das auf die Webkonsole zugegriffen werden soll (z. B. `192.0.2.1`) enthalten.

d. Wählen Sie Save.

Ihr Broker kann nun eingehende Verbindungen akzeptieren.

Java-Abhängigkeiten hinzufügen

Wenn Sie Apache Maven zum Automatisieren von Builds verwenden, fügen Sie die folgende Abhängigkeit zu Ihrer`pom.xml`Datei. Weitere Informationen zu Project Object Model-Dateien in Apache Maven finden Sie unter Einführung in die POM.

```
<dependency>
    <groupId>com.rabbitmq</groupId>
    <artifactId>amqp-client</artifactId>
    <version>5.9.0</version>
</dependency>
```

Wenn Sie Gradle zum Automatisieren von Builds verwenden, deklarieren Sie die folgende Abhängigkeit.

```
dependencies {
    compile 'com.rabbitmq:amqp-client:5.9.0'
}
```

Import **`Connection`** und **`Channel`** Klassen

Der RabbitMQ Java-Client verwendet `com.rabbitmq.client` als Paket der obersten Ebene, wobei `Connection` und `Channel` API Klassen jeweils für eine AMQP 0-9-1-Verbindung und einen Kanal stehen. Importieren Sie die `Connection` und `Channel` Klassen vor der Verwendung, wie im folgenden Beispiel gezeigt.

```
import com.rabbitmq.client.Connection;
import com.rabbitmq.client.Channel;
```

Erstellen Sie ein **`ConnectionFactory`** und verbinden Sie es mit Ihrem Broker

Mithilfe des folgenden Beispiels können Sie eine Instance der `ConnectionFactory`-Klasse mit den gegebenen Parametern. Verwenden Sie die `setHost` Methode um den Broker-Endpunkt zu konfigurieren, den Sie zuvor notiert haben. Für`AMQPS`Wire-Level-Verbindungen, Port verwenden`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();
```

Veröffentlichen einer Nachricht in einem Börse

Sie können `Channel.basicPublish` verwenden, um Nachrichten in einem Austausch veröffentlichen. Im folgenden Beispiel wird die AMQP `Builder` Klasse verwendet, um ein Nachrichteneigenschaftenobjekt mit dem Inhaltstyp zu erstellen. `plain/text`

```
byte[] messageBodyBytes = "Hello, world!".getBytes();
channel.basicPublish(exchangeName, routingKey,
             new AMQP.BasicProperties.Builder()
               .contentType("text/plain")
               .userId("userId")
               .build(),
               messageBodyBytes);
```

> **Note**
>
> Beachten Sie, dass`BasicProperties`ist eine innere Klasse der automatisch generierten Holder-Klasse,`AMQP`.

Abonnieren Sie eine Warteschlange und erhalten Sie eine Nachricht

Sie können eine Nachricht erhalten, indem Sie eine Warteschlange mit der `Consumer`-Schnittstelle implementieren. Sobald sie abonniert sind, werden Nachrichten automatisch zugestellt, sobald sie eintreffen.

Der einfachste Weg, um ein`Consumer`besteht darin, die Unterklasse`DefaultConsumer`. Ein `DefaultConsumer`-Objekt kann als Teil eines `basicConsume`-Aufrufs, um das Abonnement einzurichten, wie im folgenden Beispiel gezeigt.

```
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**
>
> Weil wir `autoAck = false` spezifizieren, ist es notwendig, Nachrichten zu bestätigen, die an die `Consumer` geliefert werden, am bequemsten in der `handleDelivery`-Methode wie im Beispiel gezeigt.

Schließen Sie Ihre Verbindung und trennen Sie vom Broker

Um die Verbindung zu Ihrem RabbitMQ-Broker zu trennen, schließen Sie sowohl den Kanal als auch die Verbindung, wie im Folgenden dargestellt.

```
channel.close();
```

```
conn.close();
```

> **Note**
>
> Weitere Informationen zur Arbeit mit der RabbitMQ Java-Clientbibliothek finden Sie im RabbitMQ Java Client Guide. API

Schritt 3: (Optional) Connect zu einer AWS Lambda Funktion herstellen

AWS Lambda kann eine Verbindung zu Ihrem Amazon MQ-Broker herstellen und Nachrichten von diesem empfangen. Wenn Sie einen Broker mit Lambda verbinden, erstellen Sie eine Ereignisquellen-Zuweisung, der Nachrichten aus einer Warteschlange liest und die Funktionsynchron. Die von Ihnen erstellte Ereignisquellenzuordnung liest Nachrichten von Ihrem Broker stapelweise und konvertiert sie in eine Lambda-Payload in Form eines Objekts. JSON

So verbinden Sie Ihren Broker mit einer Lambda Funktion

1. Fügen Sie Ihrer IAM Lambda-Funktionsausführungsrolle die folgenden Rollenberechtigungen hinzu.

   - mq: DescribeBroker
   - ec2: CreateNetworkInterface
   - ec2: DeleteNetworkInterface
   - ec2: DescribeNetworkInterfaces
   - ec2: DescribeSecurityGroups
   - ec2: DescribeSubnets
   - ec2: DescribeVpcs
   - Protokolle: CreateLogGroup
   - Protokolle: CreateLogStream
   - Protokolle: PutLogEvents
   - Verwalter von Geheimnissen: GetSecretValue

> **Note**
>
> Ohne die erforderlichen IAM Berechtigungen kann Ihre Funktion keine Datensätze aus Amazon MQ MQ-Ressourcen erfolgreich lesen.

2. (Optional) Wenn Sie einen Broker ohne öffentliche Zugänglichkeit erstellt haben, müssen Sie einen der folgenden Schritte ausführen, damit Lambda eine Verbindung zu Ihrem Broker herstellen kann:

   - Konfigurieren Sie ein NAT Gateway pro öffentlichem Subnetz. Weitere Informationen finden Sie unter Internet- und VPC Dienstzugriff für verbundene Funktionen im AWS Lambda Entwicklerhandbuch.
   - Stellen Sie mithilfe eines VPC Endpunkts eine Verbindung zwischen Ihrer Amazon Virtual Private Cloud (AmazonVPC) und Lambda her. Ihr Amazon VPC muss auch eine Verbindung zu AWS Security Token Service (AWS STS) und Secrets Manager Manager-Endpunkten herstellen. Weitere Informationen finden Sie unter Configuring Interface VPC Endpoints for Lambda im AWS Lambda Developer Guide.

3. Konfigurieren Sie Ihren Broker als EreignisquelleVerwendung für eine Lambda -Funktion unter Verwendung der AWS Management Console. Sie können den Befehl auch verwenden. `create-event-source-mapping` AWS Command Line Interface

4. Schreiben Sie Code für Ihre Lambda Funktion, um die von Ihrem Broker verbrauchten Nachrichten zu verarbeiten. Die Lambda-Payload, die von der Ereignisquellen-Zuweisung abgerufen wird, hängt vom Modultyp des Brokers ab. Im Folgenden finden Sie ein Beispiel für eine Lambda -Payload für eine Amazon MQ for RabbitMQ-Warteschlange.

> **Note**
>
> Im Beispiel ist `test` der Name der Warteschlange und `/` der Name des vorgegebenen virtuellen Hosts. Beim Empfang von Nachrichten listet die Ereignisquelle Nachrichten unter `test::/` auf.

```
{
  "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=="
```

```
        }
      ]
    }
}
```

Weitere Informationen zum Verbinden von Amazon MQ mit Lambda, zu den Optionen, die Lambda für eine Amazon-MQ-Ereignisquelle unterstützt, und zu Fehlern bei der Ereignisquellen-Zuweisung finden Sie unter Verwenden von Lambda mit Amazon MQ im AWS Lambda -Entwicklerhandbuch.

Verwalten eines Amazon MQ -Brokers

Nachdem Sie einen Broker erstellt haben, können Sie die verschiedenen Komponenten Ihres Amazon MQ-Brokers verwalten und verwalten.

Themen

Zusätzliche Amazon MQ-Broker-Einstellungen konfigurieren

Möglicherweise möchten Sie zusätzliche Einstellungen für Ihren Broker konfigurieren. Sie können während der Brokererstellung zusätzliche Einstellungen für Ihren Broker in der Konsole konfigurieren. Zusätzliche Einstellungen können Konfigurationen und VPCs öffentlichen Zugriff beinhalten.

Important

- Subnetz(e) Für den Single-Instance-Broker ist ein Subnetz erforderlich (z. B. das Standard-Subnetz). Für einen aktiv/standby-Broker sind zwei Subnetze erforderlich.
- Sicherheitsgruppe(n) Sowohl für die Single-Instance-Broker, als auch für die Aktive/Standby-Broker mit hoher Verfügbarkeit ist mindestens eine Sicherheitsgruppe erforderlich (z. B. die Standard-Sicherheitsgruppe).
- VPC— Das (die) Subnetz (e) und die Sicherheitsgruppe (n) eines Brokers müssen sich in demselben VPC befinden. EC2-Classic-Ressourcen werden nicht unterstützt. Amazon MQ unterstützt nur VPC Standard-Tenancy und keine Dedicated VPC Tenancy.

- Verschlüsselung – Wählen Sie den Kundenmasterschlüssel zum Verschlüsseln der Daten aus. Siehe Verschlüsselung im Ruhezustand.
- Öffentlicher Zugriff — Wenn Sie den öffentlichen Zugriff deaktivieren, ist der Broker nur innerhalb Ihres Unternehmens zugänglich. VPC Weitere Informationen erhalten Sie unter Broker ohne öffentlichen Zugriff bevorzugen und Zugriff auf die Amazon MQ Broker-Webkonsole ohne öffentlichen Zugriff.

1. Erweitern Sie den Abschnitt Additional settings (Erweiterte Einstellungen).
2. Wählen Sie im Abschnitt Configuration (Konfiguration) die Option Create a new configuration with default values (Neue Konfiguration mit Standardwerten erstellen) oder Select an existing configuration (Vorhandene Konfiguration auswählen) aus. Weitere Informationen finden Sie unter Amazon MQ Broker Configuration Parameters.
3. Wählen Sie im Abschnitt Logs aus, ob Sie Allgemeine Logs und Audit-Logs in Amazon CloudWatch Logs veröffentlichen möchten. Weitere Informationen finden Sie unter Monitoring and logging Amazon MQ brokers.

   Important

   Wenn Sie die -Berechtigung nicht Ihrem `CreateLogGroup` Amazon MQ-Benutzer hinzufügen, bevor der Benutzer den Broker erstellt oder neu startet, wird die Protokollgruppe nicht von Amazon MQ erstellt.
   Wenn Sie keine ressourcenbasierte Richtlinie für Amazon MQ konfigurieren, kann der Broker die Protokolle nicht in Logs veröffentlichen. CloudWatch
4. Konfigurieren Sie im Abschnitt Netzwerk und Sicherheit die Konnektivität Ihres Brokers:

   a. Führen Sie eine der folgenden Aktionen aus:

   - Wählen Sie StandardVPC, Subnetz (e) und Sicherheitsgruppe (n) verwenden aus.
   - Wählen Sie Bestehende auswählenVPC, Subnetz (n) und Sicherheitsgruppe (n) aus.

     1. Wenn Sie diese Option wählen, können Sie eine neue Virtual Private Cloud (VPC) auf der VPC Amazon-Konsole erstellen, eine bestehende VPC auswählen oder die Standardversion auswählenVPC. Weitere Informationen finden Sie unter Was ist AmazonVPC? im VPCAmazon-Benutzerhandbuch.

2. Nachdem Sie ein erstellt oder ausgewählt habenVPC, können Sie auf der VPC Amazon-Konsole neue Subnetze erstellen oder bestehende auswählen. Weitere Informationen finden Sie unter VPCsund Subnetze im VPCAmazon-Benutzerhandbuch.
3. Nach der Erstellung oder der Auswahl der Subnetze können Sie die Sicherheitsgruppe(n) auswählen.

b. Wählen Sie den Kundenhauptschlüssel (CMK) aus, der zur Verschlüsselung Ihrer Daten verwendet werden soll. Siehe Verschlüsselung im Ruhezustand.

c. Wählen Sie die Öffentliche Zugänglichkeit für Ihren Broker.

5. Konfigurieren Sie im Abschnitt Maintenance (Wartung) den Wartungszeitplan für Ihren Broker:

a. Um Upgrades auf neue Versionen Ihres Brokers vorzunehmen, wenn sie von Apache veröffentlicht werden, wählen Sie Enable automatic minor version upgrades. Automatische Upgrades erfolgen während des Wartungsfensters, das durch den Wochentag, die Tageszeit (im 24-Stunden-Format) und die Zeitzone (UTCstandardmäßig) definiert wird.

> **Note**
>
> Wenn eine der Broker-Instances einer Wartung unterzogen wird, dauert es eine kurze Zeit, bis Amazon MQ von Amazon MQ von der inaktiven Instance außer Betrieb gesetzt wurde. Auf diese Weise kann die fehlerfreie Standby-Instance aktiv werden und mit der Annahme eingehender Kommunikation beginnen.

b. Führen Sie eine der folgenden Aktionen aus:

- Um Amazon MQ die automatische Auswahl des Wartungsfensters zu erlauben, wählen Sie Keine Präferenz.
- Wenn Sie ein benutzerdefiniertes Wartungsfenster festlegen möchten, wählen Sie Select maintenance window (Wartungsfenster wählen) aus, und geben Sie anschließend den Start-Tag und die Startzeit der Upgrades ein.

Planung des Wartungsfensters für einen Amazon MQ-Broker

In regelmäßigen Abständen führt Amazon MQ während des Wartungsfensters Wartungsarbeiten an der Hardware, dem Betriebssystem oder der Engine-Software eines Message Brokers durch. Wenn Sie beispielsweise automatische Upgrades für kleinere Versionen aktiviert oder den Broker-Instance-Typ geändert haben, wendet Amazon MQ Ihre Änderungen während des nächsten geplanten

Wartungsfensters an. Die Dauer der Wartung kann je nach den für Ihren Message Broker geplanten Vorgängen bis zu zwei Stunden dauern. Sie können Ausfallzeiten während eines Wartungsfensters minimieren, indem Sie einen Broker-Bereitstellungsmodus mit hoher Verfügbarkeit in mehreren Availability Zones (AZ) auswählen.

Amazon MQ for ActiveMQ bietet Aktiv-/Standby-Bereitstellungen für hohe Verfügbarkeit. Im Aktiv-/Standby-Modus führt Amazon MQ Wartungsarbeiten für eine Instanz nach der anderen durch, und mindestens eine Instance bleibt verfügbar. Darüber hinaus können Sie ein Netzwerk von Brokern mit wöchentlichen Wartungsfenstern konfigurieren. Amazon MQ for RabbitMQ bietet die Cluster-Bereitstellungen für hohe Verfügbarkeit. Bei Cluster-Bereitstellungen führt Amazon MQ die Wartungsarbeiten an einem Knoten nach dem anderen durch, indem immer mindestens zwei Knoten ausgeführt werden.

Wenn Sie Ihren Broker zum ersten Mal erstellen, können Sie das Wartungsfenster so planen, dass es einmal pro Woche zu einer bestimmten Zeit stattfindet. Sie können das Wartungsfenster eines Brokers bis zu vier Mal vor dem nächsten geplanten Wartungsfenster anpassen. Sobald ein Broker-Wartungsfenster abgeschlossen ist, setzt Amazon MQ das Limit zurück, und Sie können den Zeitplan erneut anpassen, bevor das nächste Wartungsfenster beginnt. Die Verfügbarkeit des Brokers wird nicht beeinträchtigt, wenn das Wartungsfenster des Brokers angepasst wird.

Um das Broker-Wartungsfenster anzupassen, können Sie den AWS Management Console AWS CLI, oder den Amazon MQ API verwenden.

Planen Sie das Wartungsfenster für den Broker mithilfe der AWS Management Console

Um das Broker-Wartungsfenster anzupassen, verwenden Sie AWS Management Console

1. Melden Sie sich bei der Amazon MQ-Konsole an.
2. Wählen Sie im linken Navigationsbereich die Option Broker, un dann in der Liste den Broker aus, den Sie upgraden möchten.
3. Klicken Sie auf der Seite Broker-Details auf Edit.
4. Unter Wartung, gehen Sie für wie folgt vor.
    a. Für Start-Tag wählen Sie in der Dropdown-Liste einen Wochentag aus, z. B. Sonntag.
    b. Für Startzeit wählen Sie die Stunde und Minute des Tages aus, die Sie für das nächste Broker-Wartungsfenster planen möchten, zum Beispiel 12:00.

> **Note**
>
> Die Startzeitoptionen sind in der Zeitzone UTC +0 konfiguriert.

5. Scrollen Sie auf der Seite nach unten und klicken Sie auf Speichern. Das Wartungsfenster wird sofort eingestellt.
6. Auf der Seite mit den Broker-Details unter Maintenance window (Wartungsfenster), stellen Sie sicher, dass Ihr neuer bevorzugter Zeitplan angezeigt wird.

Planen Sie das Broker-Wartungsfenster mithilfe der AWS CLI

Um das Broker-Wartungsfenster anzupassen, verwenden Sie den AWS CLI

1. Verwenden Sie den CLI Befehl update-broker und geben Sie die folgenden Parameter an, wie im Beispiel gezeigt.

   - `--broker-id` - Die eindeutige ID, die Amazon MQ für die Broker-Instance generiert. Sie können die ID Ihres Brokers analysieren. ARN In Anbetracht der folgenden ARN Bedingungen wäre `b-1234a5b6-78cd-901e-2fgh-3i45j6k178l9` die Broker-ID beispielsweise. `arn:aws:mq:us-east-2:123456789012:broker:MyBroker:b-1234a5b6-78cd-901e-2fgh-3i45j6k178l9`
   - `--maintenance-window-start-time`— Die Parameter, die die Startzeit des wöchentlichen Wartungsfensters bestimmen, die in der folgenden Struktur angegeben ist.
     - `DayOfWeek`— Der Wochentag, in der folgenden Syntax:`MONDAY| TUESDAY | WEDNESDAY | THURSDAY | FRIDAY | SATURDAY | SUNDAY`
     - `TimeOfDay` - Die Zeit im 24-Stunden-Format.
     - `TimeZone`— (Optional) Die Zeitzone, entweder im Format Land/Stadt oder im UTC Offset-Format. Standardmäßig auf UTC festgelegt.

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

2. (Optional) Verwenden Sie den CLI Befehl describe-broker, um zu überprüfen, ob das Wartungsfenster erfolgreich aktualisiert wurde.

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

Planen Sie das Broker-Wartungsfenster mithilfe von Amazon MQ API

So passen Sie das Broker-Wartungsfenster mithilfe von Amazon MQ an API

1. Verwenden Sie die UpdateBrokerAPIOperation. Geben Sie `broker-id` an als Pfadparameter. In den folgenden Beispielen wird von einem Broker in der `us-west-2`Region ausgegangen. Weitere Informationen zu den verfügbaren Amazon-MQ-Endpunkten finden Sie unter Amazon-MQ-Endpunkte und -Kontingente in der Allgemeine AWS-Referenz

   ```
   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
   ```

   Verwenden Sie den `maintenanceWindowStartTime`-Parameter und den `WeeklyStartTime`-Ressourcentyp in der Anforderungsnutzlast.

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

2. (Optional) Verwenden Sie den DescribeBrokerAPIVorgang, um zu überprüfen, ob das Wartungsfenster erfolgreich aktualisiert wurde. `broker-id`ist als Pfadparameter angegeben.

   ```
   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
   ```

Aktualisieren einer Amazon MQ-Broker-Engine-Version

Amazon MQ stellt regelmäßig neue Broker-Engine-Versionen für alle unterstützten Broker-Engine-Typen bereit. Neue Engine-Versionen beinhalten Sicherheitspatches, Bugfixes und andere Verbesserungen der Broker-Engine.

Amazon MQ organisiert Versionsnummern gemäß der semantischen Versionsspezifikation als. `X.Y.Z` XBezeichnet in Amazon MQ MQ-Implementierungen die Hauptversion, `Y` steht für die Nebenversion und `Z` gibt die Patch-Versionsnummer an. Es gibt zwei Typen von Aktualisierungen:

- Hauptversions-Upgrade: Tritt auf, wenn sich die Versionsnummern der Haupt-Engine ändern. Beispielsweise wird ein Upgrade von Version 1.0 auf Version 2.0 als Hauptversions-Upgrade betrachtet.
- Unterversion-Upgrade: Tritt auf, wenn sich nur die Versionsnummer der Neben-Engine ändert. Zum Beispiel ein Upgrade von Version 1. 5 auf Version 1. 6 wird als geringfügiges Versionsupgrade betrachtet.

Sie können Ihren Broker jederzeit manuell auf die nächste unterstützte Haupt- oder Nebenversion aktualisieren. Wenn Sie automatische Upgrades für Nebenversionen aktivieren, aktualisiert Amazon MQ Ihren Broker auf die neueste unterstützte Patch-Version. Für alle Broker, die Engine-Version 3.13 und höher verwenden, verwaltet Amazon MQ während des Wartungsfensters Upgrades auf die neueste unterstützte Patch-Version. Amazon MQ aktualisiert Ihren Broker auf die nächste Nebenversion, wenn der Support für die aktuelle Nebenversion ausläuft. Sowohl manuelle als auch automatische Versions-Upgrades erfolgen während des geplanten Wartungsfensters oder nachdem SieSo starten Sie Ihren Broker neu.

In den folgenden Themen wird beschrieben, wie Sie die Broker-Engine-Version manuell aktualisieren und automatische Nebenversions-Upgrades aktivieren können.

Themen

- Manuelles Upgraden der Engine-Version
- Automatisches Upgraden der Engine-Unterversion

Manuelles Upgraden der Engine-Version

Um die Engine-Version eines Brokers manuell auf eine neue Haupt- oder Nebenversion zu aktualisieren, können Sie den AWS Management Console AWS CLI, oder den Amazon MQ API verwenden.

AWS Management Console

Um die Engine-Version eines Brokers zu aktualisieren, verwenden Sie AWS Management Console

1. Melden Sie sich bei der Amazon MQ-Konsole an.
2. Wählen Sie im linken Navigationsbereich die Option Broker, un dann in der Liste den Broker aus, den Sie upgraden möchten.
3. Klicken Sie auf der Seite Broker-Details auf Edit.
4. UNDERTechnische Daten, für-Broker-Engine-VersionWählen Sie in der Dropdown-Liste die neue Versionsnummer .
5. Scrollen Sie ans Seitenende und wählen SieÄnderungen im Zeitplan.
6. Klicken Sie auf derÄnderungen für Broker-Seite, fürWann Änderungen angewendet werdenWählen Sie eine der folgenden Optionen .
   - Wählen Sie Nach dem nächsten Neustart, wenn Sie möchten, dass Amazon MQ das Versions-Upgrade während des nächsten geplanten Wartungsfensters abschliesst.
   - Wählen Sie Sofort, wenn Sie den Broker neu starten und die Engine-Version sofort aktualisieren möchten.

   > **Important**
   >
   > Ihr Broker ist offline, während er neu gestartet wird.
7. Wählen Sie Anwenden, um die Anwendung der Änderungen abzuschließen.

AWS CLI

Um die Engine-Version eines Brokers zu aktualisieren, verwenden Sie AWS CLI

1. Verwenden Sie den CLI Befehl update-broker und geben Sie die folgenden Parameter an, wie im Beispiel gezeigt.

- `--broker-id` - Die eindeutige ID, die Amazon MQ für die Broker-Instance generiert. Sie können die ID Ihres Brokers analysieren. ARN In Anbetracht der folgenden ARN Bedingungen wäre `b-1234a5b6-78cd-901e-2fgh-3i45j6k178l9` die Broker-ID beispielsweise. `arn:aws:mq:us-east-2:123456789012:broker:MyBroker:b-1234a5b6-78cd-901e-2fgh-3i45j6k178l9`
- `--engine-version`— Die Engine-Versionsnummer für die Broker-Engine, auf die das Upgrade durchgeführt wird.

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

2. (Optional) Verwenden Sie den CLI Befehl reboot-broker, um Ihren Broker neu zu starten, wenn Sie die Engine-Version sofort aktualisieren möchten.

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

Wenn Sie Ihren Broker nicht neu starten und die Änderungen sofort anwenden möchten, aktualisiert Amazon MQ den Broker während des nächsten geplanten Wartungsfensters.

> **Important**
>
> Ihr Broker ist offline, während er neu gestartet wird.

Amazon MQ API

Um die Engine-Version eines Brokers mithilfe von Amazon MQ zu aktualisieren API

1. Verwenden Sie die UpdateBrokerAPIOperation. Geben Sie `broker-id` an als Pfadparameter. In den folgenden Beispielen wird von einem Broker in der `us-west-2`Region ausgegangen. Weitere Informationen zu den verfügbaren Amazon-MQ-Endpunkten finden Sie unter Amazon-MQ-Endpunkte und -Kontingente in der Allgemeine AWS-Referenz

```
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
```

Verwenden Sie `engineVersion` in der Anforderungs-Nutzlast, um die Versionsnummer für den Broker anzugeben, auf den ein Upgrade durchgeführt werden soll.

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

2. (Optional) Verwenden Sie den RebootBrokerAPIVorgang, um Ihren Broker neu zu starten, wenn Sie die Engine-Version sofort aktualisieren möchten. `broker-id`ist als Pfadparameter angegeben.

```
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
```

Wenn Sie Ihren Broker nicht neu starten und die Änderungen sofort anwenden möchten, aktualisiert Amazon MQ den Broker während des nächsten geplanten Wartungsfensters.

> **Important**
>
> Ihr Broker ist offline, während er neu gestartet wird.

Automatisches Upgraden der Engine-Unterversion

Sie können steuern, ob das automatische Nebenversions-Upgrade für einen Broker aktiviert wird, wenn Sie den Broker zum ersten Mal erstellen, oder indem Sie die Brokereinstellungen ändern. Um auto Nebenversions-Upgrades für einen vorhandenen Broker zu aktivieren, können Sie Amazon MQ AWS Management Console AWS CLI, den oder Amazon MQ API verwenden.

AWS Management Console

Um automatische Upgrades für Nebenversionen zu aktivieren, verwenden Sie AWS Management Console

1. Melden Sie sich bei der Amazon MQ-Konsole an.

2. Wählen Sie im linken Navigationsbereich die Option Broker, un dann in der Liste den Broker aus, den Sie upgraden möchten.
3. Klicken Sie auf der Seite Broker-Details auf Edit.
4. UNDERWartung, wählen SieAktivieren von automatischen Upgrades von Unterversionen.

> **Note**
>
> Wenn die Option bereits ausgewählt ist, müssen Sie keine Änderungen vornehmen.

5. Wählen Sie unten auf der Seite die Option Save aus.

AWS CLI

Um automatische Upgrades für Nebenversionen über den zu aktivieren AWS CLI, verwenden Sie den CLI Befehl update-broker und geben Sie die folgenden Parameter an.

- `--broker-id` - Die eindeutige ID, die Amazon MQ für die Broker-Instance generiert. Sie können die ID Ihres Brokers analysieren. ARN In Anbetracht der folgenden ARN Bedingungen wäre `b-1234a5b6-78cd-901e-2fgh-3i45j6k178l9` die Broker-ID beispielsweise. `arn:aws:mq:us-east-2:123456789012:broker:MyBroker:b-1234a5b6-78cd-901e-2fgh-3i45j6k178l9`
- `--auto-minor-version-upgrade`— Aktiviert die Option für automatische Unterversions-Upgrades.

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

Wenn Sie automatische Nebenversions-Upgrades für Ihren Broker deaktivieren möchten, verwenden Sie den `--no-auto-minor-version-upgrade`-Parameter.

Amazon MQ API

Verwenden Sie den UpdateBrokerAPIVorgang, um automatische Upgrades für kleinere Versionen über Amazon MQ API zu aktivieren. Geben Sie `broker-id` als Pfadparameter an. Im folgenden Beispiel wird von einem Broker in der `us-west-2`Region ausgegangen. Weitere Informationen zu den verfügbaren Amazon-MQ-Endpunkten finden Sie unter Amazon-MQ-Endpunkte und -Kontingente in der Allgemeine AWS-Referenz

```
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
```

Verwenden Sie die `autoMinorVersionUpgrade`-Eigenschaft in der Anforderungs-Nutzlast, um das automatische Nebenversions-Upgrade zu aktivieren.

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

Wenn Sie automatische Nebenversions-Upgrades für Ihren Broker deaktivieren möchten, legen Sie `"autoMinorVersionUpgrade": "false"`in der Anforderungsnutzlast fest.

Status des Amazon MQ-Brokers

Der aktuelle Zustand eines Brokers wird durch einen Status angegeben. In der folgenden Tabelle wird der Status eines Amazon MQ-Brokers aufgelistet.

Konsole	API	Beschreibung
Fehler beim Erstellen	`CREATION_FAILED`	Der Broker konnte nicht erstellt werden.
Wird erstellt	`CREATION_IN_PROGRESS`	Der Broker wird derzeit erstellt.
Wird gelöscht	`DELETION_IN_PROGRESS`	Der Broker wird derzeit gelöscht.
Laufender Neustart	`REBOOT_IN_PROGRESS`	Die Broker wird derzeit neu gestartet.
In Ausführung	`RUNNING`	Die Broker ist betriebsbereit.
Kritische Aktion erforderlich	`CRITICAL_ACTION_RE QUIRED`	Der Broker läuft, befindet sich aber in einem herunterg

Konsole	API	Beschreibung
		estuften Zustand und erfordert sofortiges Handeln. Anweisungen zur Behebung des Problems finden Sie, indem Sie den erforderl ichen Code für die Aktion aus der Liste Fehlerbehebung auswählen.

Auflisten von Amazon MQ-Brokern und Anzeigen von Broker-Details

Wenn Sie von Amazon MQ anfordern, einen Broker zu erstellen, kann der Vorgang ca. 10 Minuten dauern.

Im folgenden Beispiel wird aufgezeigt, wie Sie die Existenz Ihres Brokers bestätigen können, indem Sie Ihre Broker in der aktuellen Region mithilfe der AWS Management Console auflisten.

So listen Sie Broker auf und zeigen Broker-Details an

1. Melden Sie sich bei der Amazon MQ-Konsole an.

   Ihre Broker in der aktuellen Region werden aufgelistet.

   Für jeden Broker werden die folgenden Informationen angezeigt:

   - Name
   - Erstellungsdatum
   - Status
   - Bereitstellungsmodus
   - Instance-Typ
2. Wählen Sie den Namen Ihres Brokers.

   Für ActiveMQ-Broker auf der ***MyBroker***Auf der Seite werden die konfigurierten Details für Ihren Broker angezeigt:

Für Amazon MQ for RabbitMQ-Broker können Sie Ihre ausgewählten Einstellungen auf der ***MyBroker2***Seite, im Abschnitt Details, wie im Folgenden dargestellt.

Unter dem Abschnitt Details werden die folgenden Informationen angezeigt:

- Im Bereich Verbindungen für Amazon MQ für ActiveMQ-Broker, die Webkonsole URL und die Protokollendpunkte auf Wire-Level-Ebene.

  Im Bereich Verbindungen für Amazon MQ für RabbitMQ-Broker, die Webkonsole URL und den sicheren Endpunkt. AMQP
- Für Amazon MQ für ActiveMQ Broker finden Sie im Benutzer-Abschnitt im Benutzer im Zusammenhang mit dem Broker

> **Important**
>
> Die Verwaltung von Benutzern über den AWS Management Console und den Amazon MQ API wird für Amazon MQ for RabbitMQ-Broker nicht unterstützt.

Zugriff auf die Amazon MQ Broker-Webkonsole ohne öffentlichen Zugriff

Wenn Sie den öffentlichen Zugriff für Ihren Broker deaktivieren, müssen Sie die folgenden Schritte ausführen, um auf die Broker-Webkonsole zuzugreifen.

Voraussetzungen

Um die folgenden Schritte durchführen zu können, müssen Sie Folgendes konfigurieren:

- VPCs
  - Das VPC ohne Internet-Gateway, an das der Amazon MQ-Broker angeschlossen ist, benanntprivate-vpc.
  - Ein zweitesVPC, mit einem Internet-Gateway, benanntpublic-vpc.
  - Beide VPCs müssen verbunden sein (z. B. über VPCPeering), damit die öffentlichen EC2 Amazon-Instances mit den privaten EC2 VPC Instances kommunizieren VPC können.
  - Wenn Sie VPC Peering verwenden, VPCs müssen die Routing-Tabellen für beide für die Peering-Verbindung konfiguriert werden.

- Sicherheitsgruppen
  - Die Sicherheitsgruppe, die für die Erstellung des Amazon MQ-Brokers verwendet wird, namens `private-sg`.
  - Eine zweite Sicherheitsgruppe, die für die EC2 Instanz in der verwendet wird `public-vpc`VPC, benannte. `public-sg`
  - `private-sg` muss eingehende Verbindungen zulassen von `public-sg`. Wir empfehlen, diese Sicherheitsgruppe auf Port 8162 für ActiveMQ und auf Port 443 für RabbitMQ zu beschränken.
  - `public-sg` muss eingehende Verbindungen von Ihrem Computer auf Port 22 zulassen.

Um auf die Webkonsole eines Amazon MQ-Brokers ohne öffentlichen Zugriff zuzugreifen

1. Erstellen Sie eine EC2 Linux-Instanz in `public-vpc` (mit einer öffentlichen IP, falls erforderlich).
2. Um zu überprüfen, ob Ihre korrekt konfiguriert VPC ist, stellen Sie eine `ssh` Verbindung zur EC2 Instanz her und verwenden Sie den `curl` Befehl mit dem URI Ihres Brokers.
3. Erstellen Sie von Ihrem Computer aus einen `ssh` Tunnel zur EC2 Instanz, indem Sie den Pfad zu Ihrer privaten Schlüsseldatei und die IP-Adresse Ihrer öffentlichen EC2 Instanz verwenden. Beispielsweise:

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

   Ein Forward-Proxy-Server wird auf Ihrem Computer gestartet.
4. Installieren Sie einen Proxy-Client, z. B. FoxyProxyauf Ihrem Computer.
5. Konfigurieren Sie Ihren Proxy-Client mit den folgenden Einstellungen:
   - Geben Sie als Proxy-Typ `SOCKS5` an.
   - Geben Sie für IP-Adresse, DNS Name und Servername Folgendes an`localhost`.
   - Als Port `8080`.
   - Entfernen Sie alle vorhandenen URL Muster.
   - Geben Sie für das URL Muster Folgendes an `*.mq.*.amazonaws.com*`
   - Geben Sie als Verbindungstyp `HTTP(S)` an.

Wenn Sie Ihren Proxy-Client aktivieren, können Sie auf die Webkonsole auf Ihrem Computer zugreifen.

Neustart eines Amazon MQ-Brokers

Zur Anwendung einer neuen Konfiguration können Sie einen Broker neu starten.

> **Note**
>
> Falls Ihr ActiveMQ-Broker nicht mehr reagiert, können Sie ihn neu starten, um den Fehlerzustand zu beheben.

Das folgende Beispiel zeigt, wie Sie einen Amazon MQ-Broker mithilfe der AWS Management Console neu starten.

So starten Sie einen Amazon MQ-Broker neu

1. Melden Sie sich bei der Amazon MQ-Konsole an.
2. Wählen Sie aus der Brokerliste den Namen Ihres Brokers aus (z. B. MyBroker).
3. Auf der ***MyBroker***Wählen Sie auf der Seite Aktionen, Broker neu starten aus.

   > **Important**
   >
   > Single-Instance-Broker sind während des Neustarts offline. Cluster-Broker sind verfügbar, wobei jedoch jeweils ein Knoten neu gestartet wird.

4. Wählen Sie im Dialogfeld Broker neu starten den Eintrag Neustart aus.

   Einen Broker neu zu starten dauert etwa 5 Minuten. Wenn der Neustart Änderungen der Instance-Größe beinhaltet oder auf einem Broker mit einer hohen Warteschlangentiefe durchgeführt wird, kann der Neustart länger dauern.

Löschen eines Amazon MQ-Brokers

Wenn Sie keinen Amazon MQ-Broker verwenden (und nicht damit rechnen, ihn in naher future zu verwenden), empfiehlt es sich, ihn aus Amazon MQ zu löschen, um Ihre Kosten zu senken. AWS

Das folgende Beispiel zeigt, wie Sie einen Broker mithilfe der AWS Management Console löschen können.

Löschen eines Amazon MQ-Brokers

1. Melden Sie sich bei der Amazon-MQ-Konsole an.
2. Wählen Sie in der Brokerliste Ihren Broker aus (zum Beispiel MyBroker) und wählen Sie dann Löschen.
3. Im Bereich Löschen ***`MyBroker`***? Dialogfeld, geben Sie ein `delete` und wählen Sie dann Löschen.

   Das Löschen eines Brokers dauert ca. 15 Minuten.

Amazon MQ-Broker-Instance-Typen

Die kombinierte Beschreibung der Broker-Instanceclass(`m5`,`t3`) undsize(`large`,`micro`) ist einInstance-Typ(z. B.`mq.m5.large`). In der folgenden Tabelle werden die verfügbaren Amazon MQ Broker-Instance-Typen für jeden unterstützten Engine-Typ aufgeführt.

Themen

- Amazon MQ für ActiveMQ Instance-Typen
- Instance-Typen von Amazon MQ für RabbitMQ

Amazon MQ für ActiveMQ Instance-Typen

> **Important**
>
> Sie können Amazon EBS nur mit der `mq.m5` Broker-Instance-Typfamilie verwenden. Weitere Informationen finden Sie unter Storage.

Instance-Typ	v CPU	Arbeitsspeicher (GiB)	Netzwerkleistung	Empfohlene Verwendung
`mq.t2.micro`	1	1	Niedrig	Bewertung
`mq.t3.micro`	2	1	Niedrig	Bewertung
`mq.m4.large`	2	8	Mittel	Produktion
`mq.m5.large`	2	8	Hoch	Produktion
`mq.m5.xlarge`	4	16	Hoch	Produktion
`mq.m5.2xlarge`	8	32	Hoch	
`mq.m5.4xlarge`	16	64	Hoch	

Weitere Informationen zu den Durchsatz betreffenden Faktoren finden Sie unter Auswählen des richtigen Broker-Instance-Typs für den besten Durchsatz.

Instance-Typen von Amazon MQ für RabbitMQ

> **Important**
>
> Sie können einen Broker nicht von einem `mq.m5.`-Instance-Typ zu einem `mq.t3.micro`-Instance-Typ herunterstufen.

Instance-Typ	v CPU	Arbeitsspeicher (GiB)	Netzwerkleistung	Anwendungsfall
`mq.t3.micro`	2	1	Niedrig	Bewertung

Instance-Typ	v CPU	Arbeitsspeicher (GiB)	Netzwerkleistung	Anwendungsfall
				Important Der `mq.t3.micro` - Instance -Typ unterstützt die Cluster-Bereitstellung nicht.
`mq.m5.large`	2	8	Hoch	Produktion
`mq.m5.xlarge`	4	16	Hoch	Produktion
`mq.m5.2xlarge`	8	32	Hoch	
`mq.m5.4xlarge`	16	64	Hoch	

Hinzufügen von Tags zu Amazon MQ MQ-Ressourcen

Zur Organisation und Identifizierung Ihrer Amazon MQ-Ressourcen für die Kostenzuordnung können Sie Metadaten-Tags hinzufügen, die den Zweck eines Brokers oder einer Konfiguration identifizieren. Dies ist besonders nützlich, wenn Sie viele Broker haben. Mithilfe von Tags zur Kostenzuweisung können Sie Ihre AWS Rechnung so organisieren, dass sie Ihrer eigenen Kostenstruktur entspricht. Melden Sie sich dazu an, um Ihre AWS Kontorechnung mit den Tagschlüsseln und -werten zu erhalten. Weitere Informationen finden Sie unter Einrichten Ihres monatlichen Kostenzuordnungsberichts im AWS Billing Benutzerhandbuch.

Beispielsweise können Sie Tags hinzufügen, die die Kostenstelle und den Zweck Ihrer Amazon MQ-Ressourcen repräsentieren:

Ressource	Schlüssel	Wert
Broker1	Cost Center	34567
	Stack	Production
Broker2	Cost Center	34567
	Stack	Production
Broker3	Cost Center	12345
	Stack	Development

Mit diesem Kennzeichnungsschema können Sie zwei Broker, die verwandte Aufgaben ausführen, in derselben Kostenstelle zusammenfassen, während Sie einen nicht verwandten Broker mit einer anderen Kostenverteilungskennzeichnung versehen.

Hinzufügen von Tags in der Amazon MQ MQ-Konsole

Sie können den Ressourcen, die Sie in der Amazon MQ MQ-Konsole erstellen, schnell Tags hinzufügen, indem Sie die folgenden Schritte ausführen:

1. Wählen Sie auf der Seite Create a broker (Broker erstellen) Additional settings (Zusätzliche Einstellungen) aus.
2. Wählen Sie unter Tags Add tag (Tag hinzufügen) aus.
3. Geben Sie ein Key (Schlüssel)- und Value (Wert)-Paar ein.
4. (Optional) Wählen Sie Add tag (Tag hinzufügen) aus, um Ihrem Broker mehrere Tags hinzuzufügen.
5. Wählen Sie Create broker (Broker erstellen) aus.

So fügen Sie beim Erstellen einer Konfiguration Tags hinzu:

1. Wählen Sie auf der Seite Create configuration (Konfiguration erstellen) Advanced (Erweitert) aus.

2. Wählen Sie unterTagsauf der Seite Erstellen einer KonfigurationTag hinzufügenaus.
3. Geben Sie ein Key (Schlüssel)- und Value (Wert)-Paar ein.
4. (Optional) Wählen Sie Add tag (Tag hinzufügen) aus, um Ihrer Konfiguration mehrere Tags hinzuzufügen.
5. Wählen Sie Create configuration (Konfiguration erstellen) aus.

Nachdem Sie Tags hinzugefügt haben, können Sie die Tags für Ihre Ressourcen in der Amazon MQ MQ-Konsole anzeigen, bearbeiten und entfernen. Sie können die Tags Ihrer Ressourcen auch mit dem REST API anzeigen. Weitere Informationen finden Sie in der Amazon MQ REST API MQ-Referenz.

Amazon MQ für ActiveMQ verwenden

Mit Amazon MQ ist es ganz einfach, einen Message Broker mit den Computing- und Speicherressourcen zu erstellen, die Ihren Anforderungen entsprechen. Sie können Broker mit dem, Amazon MQ REST API oder dem erstellen AWS Management Console, verwalten und löschen. AWS Command Line Interface

Amazon MQ für ActiveMQ-Broker kann als Single-Instance-Broker oder als Active/Standby-Broker eingesetzt werden. Für beide Bereitstellungsmodi bietet Amazon MQ eine hohe Haltbarkeit, indem seine Daten redundant gespeichert werden.

> **Note**
>
> Amazon MQ verwendetApache KahaDBals Datenspeicher. Andere Datenspeicher wie JDBC LevelDB werden nicht unterstützt.

Sie können auf Ihre Broker zugreifen, indem Sie jede Programmiersprache verwenden, die ActiveMQ unterstützt, und indem Sie sie TLS explizit für die folgenden Protokolle aktivieren:

- AMQP
- MQTT
- MQTTüber WebSocket
- OpenWire
- STOMP
- STOMPüber WebSocket

Weitere Informationen zu Amazon MQ REST APIs finden Sie in der Amazon MQ MQ-Referenz REST API.

Amazon MQ für ActiveMQ-Broker

Was ist ein Amazon MQ for ActiveMQ-Broker?

Ein Broker ist eine Message-Broker-Umgebung, die auf Amazon MQ ausgeführt wird. Dies ist der Grundblock für Amazon MQ. Die kombinierte Beschreibung der Broker-Instanceclass(`m5`,`t3`)

undsize(`large`,`micro`) ist einBroker-Instance-Typ(zum Beispiel`mq.m5.large`). Weitere Informationen finden Sie unter Broker instance types.

- Ein Single-Instance-Broker besteht aus einem Broker in einer Availability Zone. Der Broker kommuniziert mit Ihrer Anwendung und mit einem Amazon- EBS oder EFS Amazon-Speichervolumen.
- Ein aktiv/standby-Broker besteht aus zwei Brokern in zwei verschiedenen Availability Zones, die in einem redundanten Paar konfiguriert sind. Diese Broker kommunizieren synchron mit Ihrer Anwendung und mit AmazonEFS.

Weitere Informationen finden Sie unter Bereitstellungsoptionen für Amazon MQ für ActiveMQ-Broker.

Sie können automatische Upgrades auf Unterversionen aktivieren, damit Upgrades auf neue Unterversionen der Broker-Engine ausgeführt werden, sobald Apache neue Versionen veröffentlicht. Automatische Upgrades erfolgen während des Wartungsfensters, das durch den Wochentag, die Tageszeit (im 24-Stunden-Format) und die Zeitzone (UTCstandardmäßig) definiert wird.

Weitere Informationen zum Erstellen und Verwalten von Brokern finden Sie unter:

- Erste Schritte: Einen ActiveMQ-Broker erstellen und eine Verbindung zu ihm herstellen
- Broker
- Broker statuses

Unterstützte Wire-Level-Protokolle

Sie können auf Ihre Broker zugreifen, indem Sie jede Programmiersprache verwenden, die ActiveMQ unterstützt, und indem Sie sie TLS explizit für die folgenden Protokolle aktivieren:

- AMQP
- MQTT
- MQTTüber WebSocket
- OpenWire
- STOMP
- STOMPüber WebSocket

Attribute

Ein ActiveMQ-Broker verfügt über mehrere Attribute, z. B.:

- Einen Namen (`MyBroker`)
- Eine ID (`b-1234a5b6-78cd-901e-2fgh-3i45j6k178l9`)
- Ein Amazon-Ressourcenname (ARN) (`arn:aws:mq:us-east-2:123456789012:broker:MyBroker:b-1234a5b6-78cd-901e-2fgh-3i45j6k178l9`)
- Eine ActiveMQ-Webkonsole URL () `https://b-1234a5b6-78cd-901e-2fgh-3i45j6k178l9-1.mq.us-east-2.amazonaws.com:8162`

  Weitere Informationen finden Sie unter Web-Konsole in der Apache ActiveMQ-Dokumentation.

  > **Important**
  >
  > Wenn Sie eine Autorisierungszuweisung angeben, die die`activemq-webconsole`können Sie die ActiveMQ Webkonsole nicht verwenden, da die Gruppe nicht berechtigt ist, Nachrichten an den Amazon MQ -Broker zu senden oder von ihm Nachrichten zu empfangen.

- Wire-Level-Protokoll-Endpunkte:
  - `amqp+ssl://b-1234a5b6-78cd-901e-2fgh-3i45j6k178l9-1.mq.us-east-2.amazonaws.com:5671`
  - `mqtt+ssl://b-1234a5b6-78cd-901e-2fgh-3i45j6k178l9-1.mq.us-east-2.amazonaws.com:8883`
  - `ssl://b-1234a5b6-78cd-901e-2fgh-3i45j6k178l9-1.mq.us-east-2.amazonaws.com:61617`

    > **Note**
    >
    > Dies ist ein Endpunkt OpenWire .

  - `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`

Weitere Informationen finden Sie unter Transport-Konfiguration in der Apache ActiveMQ-Dokumentation.

> **Note**
>
> Für einen Aktiv-/Standby-Broker bietet Amazon MQ zwei ActiveMQ-Web-KonsolenURLs, von denen jedoch jeweils nur eine aktiv URL ist. Ebenso stellt Amazon MQ zwei Endpunkte für jedes Wire-Level-Protokoll bereit, jedoch ist jeweils nur ein Endpunkt in jedem Paar aktiv. Die `-1`- und `-2`-Suffixe bezeichnen ein redundantes Paar.

Eine vollständige Liste der Broker-Attribute finden Sie im Folgenden in der Amazon MQ REST API MQ-Referenz:

- RESTVorgangs-ID: Makler
- RESTVorgangs-ID: Makler
- RESTVorgangs-ID: Neustart des Brokers

Broker-Benutzer

Ein ActiveMQBenutzerist eine Person oder eine Anwendung, die auf die Warteschlangen und Themen eines ActiveMQ -Brokers zugreifen kann. Sie können Benutzer so konfigurieren, dass sie bestimmte Berechtigungen haben. Beispielsweise können Sie einigen Benutzern erlauben, auf dieActiveMQ-Webkonsole zuzugreifen.

Eine Gruppeist ein semantisches Label. Sie können einem Benutzer eine Gruppe zuweisen und Berechtigungen für Gruppen zum Senden, Empfangen von und Verwalten bestimmter Warteschlangen und Themen konfigurieren.

> **Important**
>
> Das Vornehmen von Änderungen an einem Benutzer wendet nicht sofort die Änderungen auf den Benutzer an. Um Ihre Änderungen zu übernehmen, müssen Sie auf den nächsten Wartungszeitraum warten oder den Broker neu starten.

Weitere Informationen zu Benutzern und Gruppen finden Sie in der folgenden Dokumentation zu Apache ActiveMQ:

- Autorisierung
- Autorisierungsbeispiel

Weitere Informationen zum Erstellen, Bearbeiten und Löschen von ActiveMQ-Benutzern finden Sie unter:

- Einen ActiveMQ-Broker-Benutzer erstellen
- Benutzer

Benutzerattribute

Eine vollständige Liste der Benutzerattribute finden Sie im Folgenden in der Amazon MQ REST API MQ-Referenz:

- RESTVorgangs-ID: Benutzer
- RESTVorgangs-ID: Benutzer

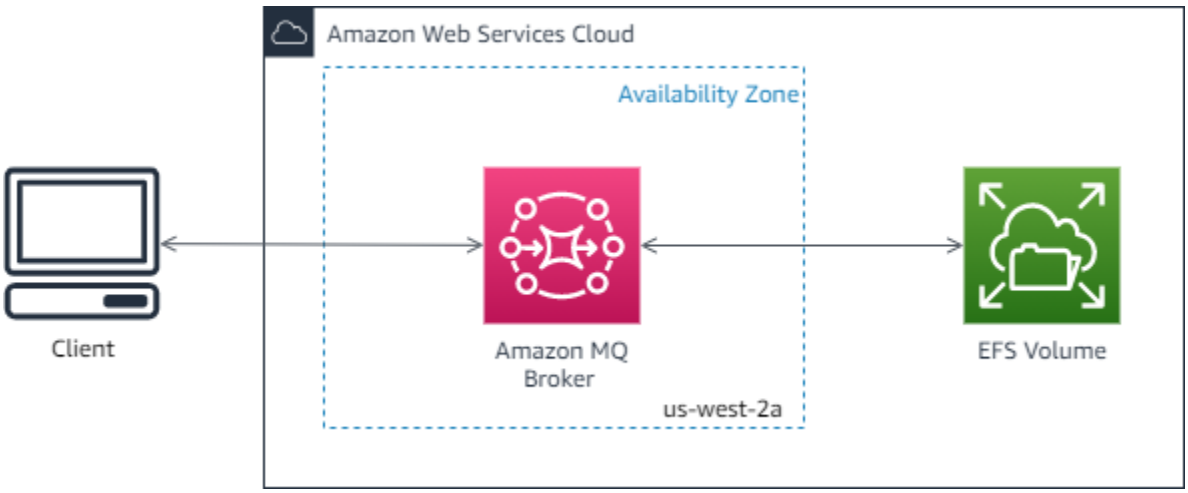
Bereitstellungsoptionen für Amazon MQ für ActiveMQ-Broker

Amazon MQ bietet Einzelinstanz- und Cluster-Bereitstellungsoptionen für Broker.

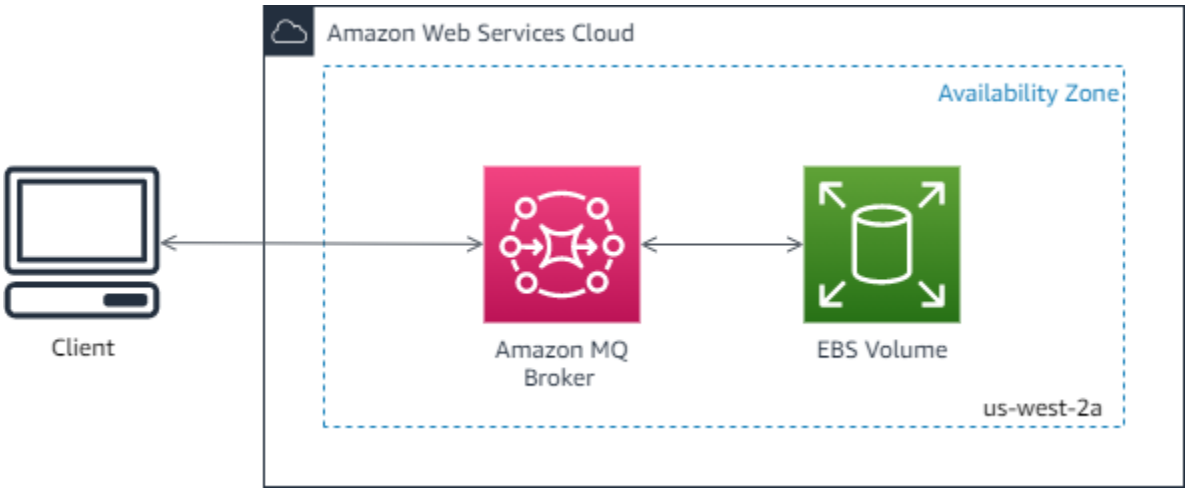
Option 1: Amazon MQ-Broker mit einer einzigen Instanz

Ein Single-Instance-Broker besteht aus einem Broker in einer Availability Zone. Der Broker kommuniziert mit Ihrer Anwendung und mit einem Amazon- EBS oder EFS Amazon-Speichervolumen. EFSAmazon-Speichervolumes sind so konzipiert, dass sie ein Höchstmaß an Haltbarkeit und Verfügbarkeit bieten, indem Daten redundant in mehreren Availability Zones (AZs) gespeichert werden. Amazon EBS bietet Speicher auf Blockebene, der für niedrige Latenz und hohen Durchsatz optimiert ist. Weitere Informationen zu Speicherungsoptionen finden Sie unter Storage.

Das folgende Diagramm zeigt einen Single-Instance-Broker, bei dem EFS Amazon-Speicher auf mehrere repliziert wird. AZs



Das folgende Diagramm zeigt einen Single-Instance-Broker mit EBS Amazon-Speicher, der auf mehrere Server innerhalb einer einzigen AZ repliziert wird.



Option 2: Amazon MQ Active/Standby-Broker für hohe Verfügbarkeit

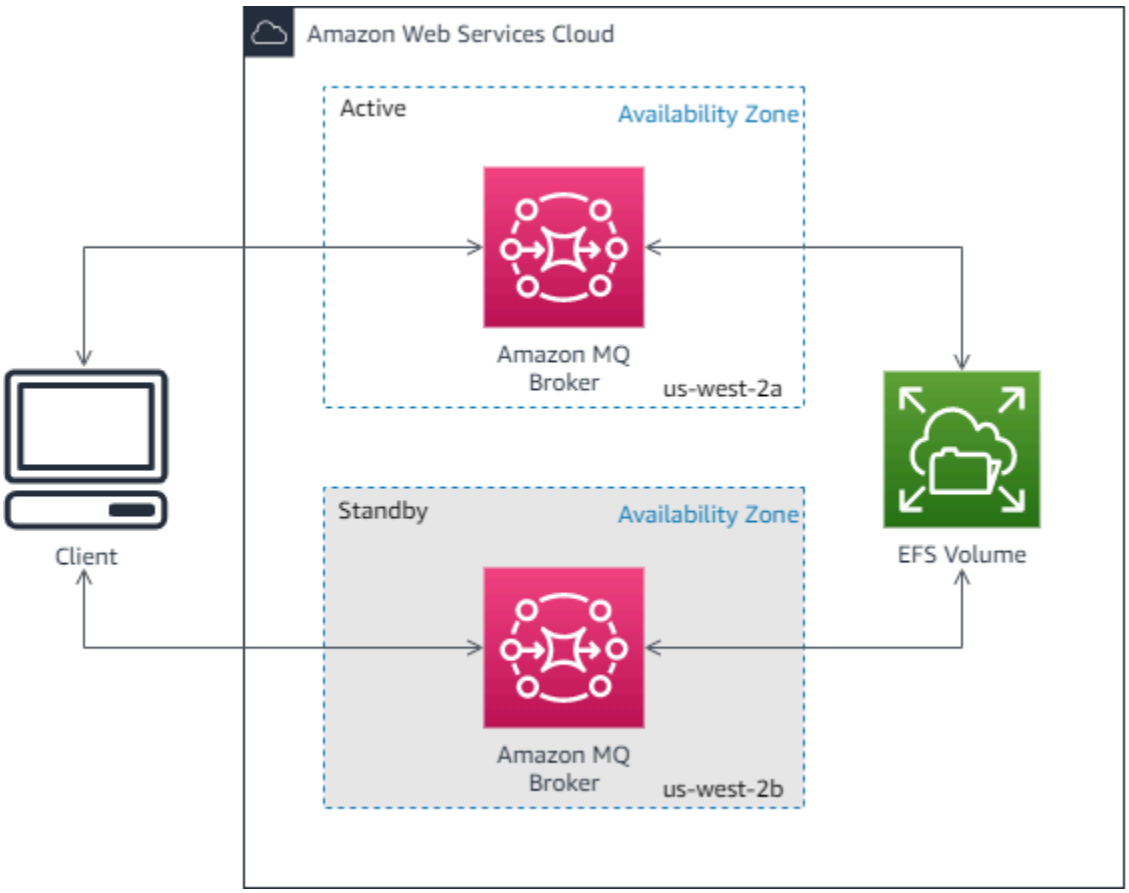
Ein aktiv/standby-Broker besteht aus zwei Brokern in zwei verschiedenen Availability Zones, die in einem redundanten Paar konfiguriert sind. Diese Broker kommunizieren synchron mit Ihrer Anwendung und mit AmazonEFS. EFSAmazon-Speichervolumes sind so konzipiert, dass sie ein Höchstmaß an Haltbarkeit und Verfügbarkeit bieten, indem Daten redundant in mehreren Availability Zones (AZs) gespeichert werden. Weitere Informationen finden Sie unter Storage.

Normalerweise ist nur jeweils eine der Broker-Instances aktiv, während sich die anderen Broker-Instances im Standby-Modus befinden. Wenn eine der Broker-Instances eine Fehlfunktion aufweist oder einer Wartung unterzogen wird, dauert es eine kurze Zeit, bis Amazon MQ die inaktive Instance von außer Betrieb gesetzt hat. Auf diese Weise kann die fehlerfreie Standby-Instance aktiv werden

und mit der Annahme eingehender Kommunikation beginnen. Wenn Sie einen Broker neu starten, dauert das Failover nur wenige Sekunden.

Für einen Aktiv-/Standby-Broker bietet Amazon MQ zwei ActiveMQ-Web-KonsolenURLs, von denen jedoch jeweils nur eine aktiv URL ist. Ebenso stellt Amazon MQ zwei Endpunkte für jedes Wire-Level-Protokoll bereit, jedoch ist jeweils nur ein Endpunkt in jedem Paar aktiv. Die `-1` und `-2` Suffixe bezeichnen ein redundantes Paar. Für Drahtebene Protokollendpunkte können Sie zulassen, dass Ihre Anwendung eine Verbindung zu einem beliebigen Endpunkt herstellen kann, indem Sie die Failover-Transport verwenden.

Das folgende Diagramm zeigt einen Aktiv-/Standby-Broker, bei dem EFS Amazon-Speicher über mehrere repliziert wird. AZs



Amazon MQ Brokernetzwerk

Amazon MQ unterstützt die ActiveMQ-Funktion für Netzwerke von Brokern.

Ein -Netzwerk von Brokernbesteht aus mehreren gleichzeitig aktivenSingle-Instance-Brokernoderaktiven/Standby-Brokern. Sie können Brokernetzwerke in einer Vielzahl von Topologien

(z. B. Concentrator hub-and-spokes, Tree oder Mesh) konfigurieren, je nach den Anforderungen Ihrer Anwendung, z. B. Hochverfügbarkeit und Skalierbarkeit. Zum Beispiel kann ein Hub-und-Spoke-Netzwerk von Brokern kann die Ausfallsicherheit erhöhen und Nachrichten erhalten, wenn ein Broker nicht erreichbar ist. Ein Netzwerk von Brokern mit einemKonzentratorTopologie kann Nachrichten von einer größeren Anzahl von Brokern sammeln, die eingehende Nachrichten akzeptieren, und sie auf zentralere Broker konzentrieren, um die Belastung vieler eingehender Nachrichten besser zu bewältigen.

Ein Tutorial und detaillierte Konfigurationsinformationen finden Sie im Folgenden:

- Creating and Configuring a Network of Brokers
- Korrekte Konfiguration Ihres Netzwerk von Brokern
- `networkConnector`
- `networkConnectionStartAsynchron`
- Netzwerke von Brokern in der ActiveMQ-Dokumentation

Vorteile aus der Nutzung eines Netzwerks von Brokern:

- Die Einrichtung eines Netzwerks von Brokern ermöglicht es Ihnen, Ihren Gesamtdurchsatz und die maximale Anzahl der Produzenten- und Konsumentenverbindungen durch Hinzufügen von Broker-Instances zu erhöhen.
- Sie können eine bessere Verfügbarkeit sicherstellen, indem Sie Ihren Produzenten und Verbrauchern ermöglichen, sich über mehrere aktive Broker-Instances zu informieren. Dies ermöglicht es ihnen, sich wieder mit einer neuen Instance zu verbinden, wenn diejenige, mit der sie gerade verbunden sind, nicht verfügbar ist.
- Da Produzenten und Verbraucher sofort wieder eine Verbindung zu einem anderen Knoten im Netzwerk der Broker herstellen können und es nicht notwendig ist, darauf zu warten, dass eine Standby-Broker-Instance befördert wird, ist die Wiederverbindung des Kunden innerhalb eines Netzwerks von Brokern schneller als bei einem aktiven/Standby-Broker für hohe Verfügbarkeit.

Themen

- Wie funktioniert ein Netzwerk von Brokern?
- Wie geht ein Netzwerk von Brokern mit Anmeldeinformationen um?
- Beispiel-Vorlagen
- Topologien für Netzwerke von Brokern

- Regionsübergreifend
- Dynamisches Failover mit Transport Connectors

Wie funktioniert ein Netzwerk von Brokern?

Amazon MQ unterstützt das ActiveMQ-Netzwerk von Brokern auf verschiedene Weise. Erstens können Sie die Parameter innerhalb der Konfiguration jedes Brokers bearbeiten, um ein Netzwerk von Brokern zu erstellen, genau wie bei nativem ActiveMQ. Zweitens verfügt Amazon MQ über Musterpläne, mit denen AWS CloudFormation der Aufbau eines Brokernetzwerks automatisiert wird. Sie können diese Beispiel-Vorlagen direkt aus der Holen Sie sich die Amazon MQ-Konsole verwenden, oder Sie können die zugehörigen AWS CloudFormation - Vorlagen bearbeiten, um eigene Topologien und Konfigurationen zu erstellen.

Ein Netzwerk von Brokern wird aufgebaut, indem ein Broker über Netzwerk-Connectors mit einem anderen verbunden wird. Sobald sie verbunden sind, bieten diese Broker eine Nachrichtenweiterleitung an. Zum Beispiel, wenn Broker1 einen Netzwerk-Connector nach Broker2 einrichtet, werden Nachrichten auf Broker1 an Broker2 weitergeleitet, falls ein Verbraucher auf diesem Broker für die Warteschlange oder das Thema vorhanden ist. Wenn der Netzwerk-Connector als `duplex` konfiguriert ist, werden Nachrichten auch von Broker2 an Broker1 weitergeleitet. Netzwerk-Connectors werden in der Configuration (Konfiguration) des Brokers konfiguriert. Siehe Amazon MQ Broker Configuration Parameters. Als Beispiel hier ein Beispieleintrag für einen `networkConnector` in einer Broker-Konfiguration:

```
<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>
```

Ein Netzwerk von Brokern stellt sicher, dass Nachrichten von einer Brokerinstance zur anderen fließen, und leitet Nachrichten nur an die Brokerinstances weiter, die entsprechende Konsumenten haben. Zum Nutzen von Brokerinstances, die im Netzwerk nebeneinander liegen, sendet ActiveMQ Nachrichten an Beratungsthemen über Produzenten und Verbraucher, die Verbindungen mit dem Netzwerk herstellen und trennen. Wenn eine Broker-Instance Informationen über einen Verbraucher erhält, der von einem bestimmten Ziel konsumiert, beginnt die Broker-Instance, Nachrichten weiterzuleiten. Weitere Informationen finden Sie unter Advisory Topics in der ActiveMQ-Dokumentation.

Wie geht ein Netzwerk von Brokern mit Anmeldeinformationen um?

Damit sich Broker A mit Broker B in einem Netzwerk verbinden kann, muss Broker A gültige Anmeldeinformationen verwenden, wie jeder andere Produzent oder Verbraucher. Anstatt ein Passwort in der `<networkConnector>`-Konfiguration von Broker A anzugeben, müssen Sie zunächst einen Benutzer auf dem Broker A mit den gleichen Werten wie ein anderer Benutzer auf dem Broker B anlegen (dies sind separate, einzigartige Benutzer, die die gleichen Werte für Benutzername und Passwort verwenden). Wenn Sie das Attribut `userName` in der `<networkConnector>`-Konfiguration angeben, fügt Amazon MQ das Passwort zur Laufzeit automatisch hinzu.

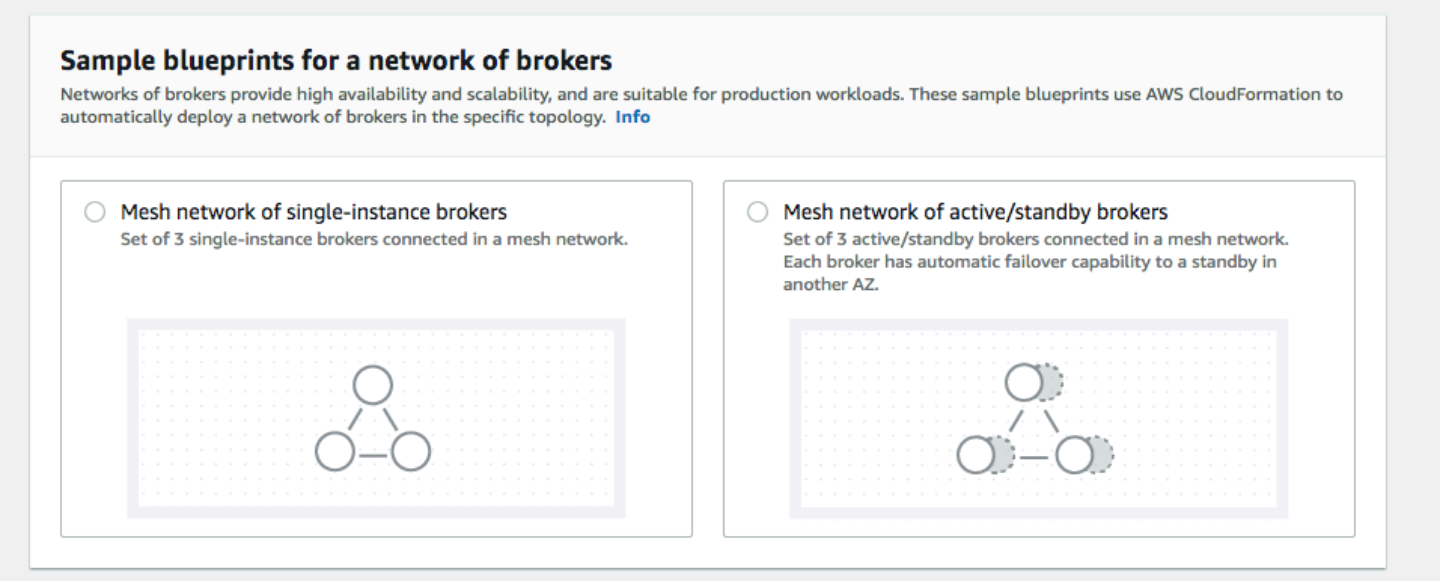
> **Important**
>
> Geben Sie kein `password`-Attribut für das `<networkConnector>` an. Wir empfehlen nicht, Klartext-Passwörter in Broker-Konfigurationsdateien zu speichern, da dadurch die Passwörter in der Holen Sie sich die Amazon MQ-Konsole sichtbar werden. Weitere Informationen finden Sie unter Configure Network Connectors for Your Broker.

Makler müssen sich im selben VPC oder im Peering-Modus befinden. VPCs Weitere Informationen finden Sie unter Voraussetzungen im Creating and Configuring a Network of Brokers Tutorial.

Beispiel-Vorlagen

Um mit der Nutzung eines Netzwerks von Brokern zu beginnen, bietet Amazon MQ Beispiel-Vorlagen. Mit diesen Beispiel-Blueprints wird eine Network of Broker-Bereitstellung und alle zugehörigen Ressourcen mithilfe von, erstellt. AWS CloudFormation Die beiden verfügbaren Beispiel-Vorlagen sind:

1. Mesh-Netzwerk von Single-Instance-Brokern
2. Mesh-Netzwerk von aktiven/Standby-Brokern



Wählen Sie auf der Seite Create brokers (Broker erstellen) eine der Beispiel-Vorlagen aus und klicken Sie auf Next (Weiter). Nachdem die Ressourcen erstellt wurden, überprüfen Sie die generierten Broker und ihre Konfigurationen in der Amazon MQ-Konsole.

Durch die Erstellung von Brokern und die Konfiguration verschiedener `networkConnector`-Elemente in den Broker-Konfigurationen können Sie ein Netzwerk von Brokern in vielen verschiedenen Topologien erstellen. Weitere Informationen zur Konfiguration eines Netzwerks von Brokern finden Sie unter Networks of Brokers (Netzwerke von Brokern) in der ActiveMQ-Dokumentation.

Topologien für Netzwerke von Brokern

Durch die Bereitstellung von Brokern und die anschließende Konfiguration von `networkConnector`-Einträgen in ihrer Konfiguration können Sie ein Netzwerk von Brokern mit unterschiedlichen Netzwerktopologien aufbauen. Ein Netzwerk-Connector ermöglicht die On-Demand-Weiterleitung von Nachrichten zwischen verbundenen Brokern. Verbindungen können als Duplex konfiguriert werden, wobei Nachrichten in beide Richtungen zwischen Brokern weitergeleitet werden, oder nicht als Duplex, wobei sich die Weiterleitung nur von einem Broker zum anderen erstreckt. Beispiel: Bei einer Duplex-Verbindung zwischen Broker1 und Broker2 werden Nachrichten von jedem Broker an den anderen weitergeleitet, falls ein Konsument vorhanden ist.

Mit einem Duplex-Netzwerk-Connector werden Nachrichten von jedem Broker an den anderen weitergeleitet. Diese werden On-Demand weitergeleitet: Wenn bei Broker2 ein Konsument für eine Nachricht auf Broker1 vorhanden ist, wird die Nachricht weitergeleitet. Ähnlich verhält es sich, wenn es auf Broker1 einen Konsumenten für eine Nachricht auf Broker2 gibt, wird die Nachricht ebenfalls weitergeleitet.

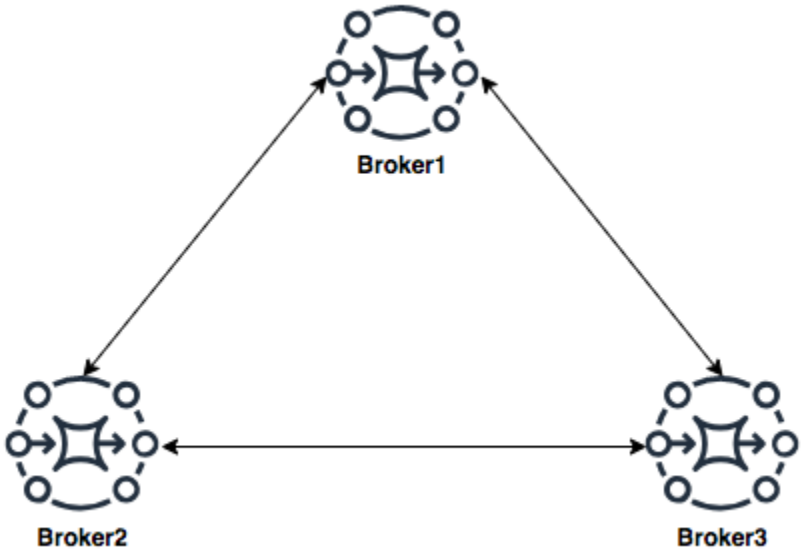
Bei Nicht-Duplex-Verbindungen werden Nachrichten nur von einem Broker zum anderen weitergeleitet. Ist in diesem Beispiel auf Broker2 ein Konsument für eine Nachricht auf Broker1 vorhanden, wird die Nachricht weitergeleitet. Nachrichten werden aber nicht von Broker2 an Broker1 weitergeleitet.

Durch die Verwendung von Duplex- und Nicht-Duplex-Netzwerk-Connectors ist es möglich, ein Netzwerk von Brokern in einer beliebigen Anzahl von Netzwerktopologien aufzubauen.

> **Note**
>
> In jedem der Beispiele für die Netzwerktopologie verweisen die `networkConnector`-Elemente auf den Endpunkt der Broker, mit denen sie sich verbinden. Ersetzen Sie die Broker-Endpunkteinträge in den `uri`-Attributen mit den Endpunkten Ihrer Broker. Siehe Listing brokers and viewing broker details.

Mesh-Topologie

Eine Mesh-Topologie bietet mehrere Broker, die alle miteinander verbunden sind. Dieses einfache Beispiel verbindet drei Single-Instance-Broker, aber Sie können mehr Broker als Mesh konfigurieren.



Diese Topologie sowie eine Topologie, die ein Mesh-Netzwerk mit aktiven/Standby-Broker-Paaren enthält, lassen sich mit Beispiel-Vorlagen in der Amazon MQ-Konsole erstellen. Sie können die Bereitstellung dieser Beispiel-Vorlagen erstellen, um ein funktionierendes Netzwerk von Brokern zu sehen und zu überprüfen, wie sie konfiguriert sind.

Sie können ein Drei-Broker-Mesh-Netzwerk wie folgt konfigurieren, indem Sie einen Netzwerk-Connector hinzufügen zu Broker1, der Duplexverbindungen zu Broker2 und zu Broker3 und eine einzige Duplexverbindung zwischen Broker2 und Broker3 herstellt.

Netzwerk-Connectors für Broker1:

```
<networkConnectors>
    <networkConnector name="connector_1_to_2" userName="myCommonUser" duplex="true"
        uri="static:(ssl://b-9876l5k4-32ji-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>
```

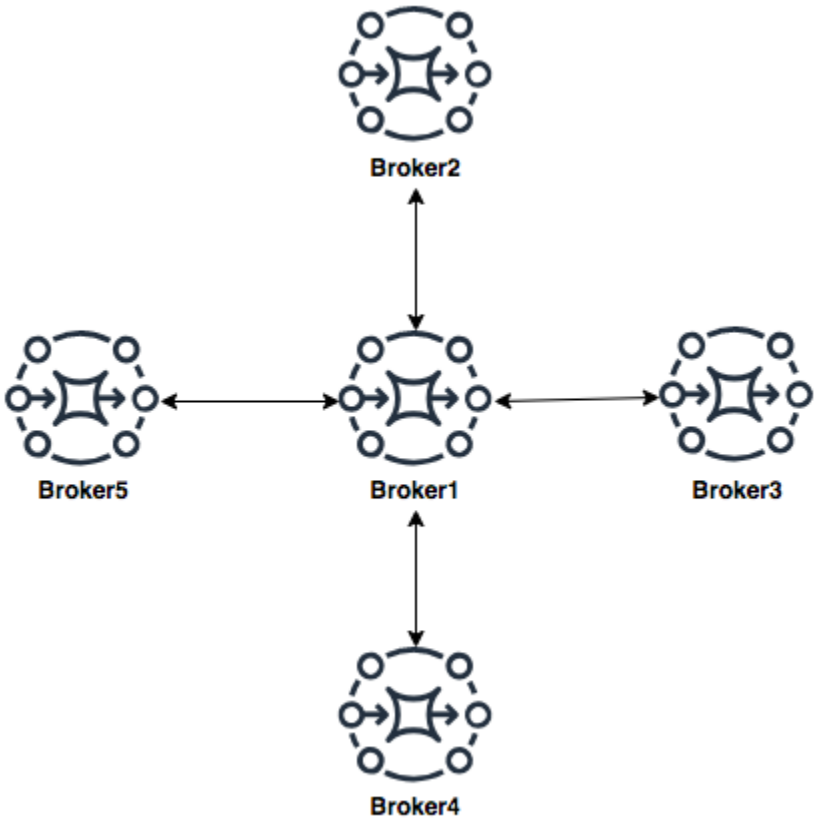
Netzwerk-Connectors für Broker2:

```
<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>
```

Durch Hinzufügen der oben genannten Connectors zu den Konfigurationen von Broker1 und Broker2 können Sie ein Mesh-Netzwerk zwischen diesen drei Brokern erstellen, das die Nachricht On-Demand zwischen allen Brokern weiterleitet. Weitere Informationen finden Sie unter Amazon MQ Broker Configuration Parameters.

Hub-and-Spoke-Topologie

In einer Hub-and-Spoke-Topologie werden Nachrichten gespeichert, wenn es zu einer Unterbrechung für einen Broker auf einem Spoke kommt. Nachrichten werden durchgehend weitergeleitet, und nur der zentrale Broker1 ist kritisch für den Betrieb des Netzwerks.

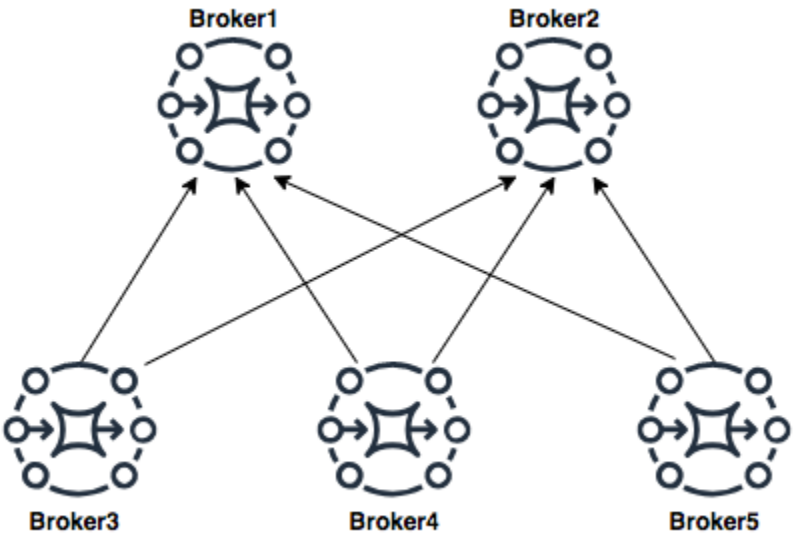


Um in diesem Beispiel das Hub-and-Spoke-Netzwerk von Brokern zu konfigurieren, können Sie einen `networkConnector` an jeden der Broker auf den Spokes in der Konfiguration von Broker1 hinzufügen.

```
<networkConnectors>
    <networkConnector name="connector_hub_and_spoke_2" userName="myCommonUser"
 duplex="true"
        uri="static:(ssl://b-1234a5b6-78cd-901e-2fgh-3i45j6k178l9-1.mq.us-
east-2.amazonaws.com:61617)"/>
    <networkConnector name="connector_hub_and_spoke_3" userName="myCommonUser"
 duplex="true"
        uri="static:(ssl://b-9876l5k4-32ji-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>
```

Konzentrator-Topologie

In dieser Beispieltopologie können die drei Broker am unteren Rand eine große Anzahl von Verbindungen verwalten, und diese Nachrichten sind auf Broker1 und Broker2 konzentriert. Jeder der anderen Broker hat eine nicht-duplexe Verbindung zu den zentraleren Brokern. Um die Kapazität dieser Topologie zu skalieren, können Sie weitere Broker hinzufügen, die Nachrichten empfangen, und diese Nachrichten in Broker1 und Broker2 konzentrieren.



Um diese Topologie zu konfigurieren, würde jeder der Broker auf der Unterseite einen Netzwerk-Connector zu jedem der Broker enthalten, auf die sie Nachrichten konzentrieren.

Netzwerk-Connectors für Broker3:

```
<networkConnectors>
    <networkConnector name="3_to_1" userName="myCommonUser" duplex="false"
        uri="static:(ssl://b-1234a5b6-78cd-901e-2fgh-3i45j6k178l9-1.mq.us-
east-2.amazonaws.com:61617)"/>
    <networkConnector name="3_to_2" userName="myCommonUser" duplex="false"
        uri="static:(ssl://b-9876l5k4-32ji-109h-8gfe-7d65c4b132a1-2.mq.us-
east-2.amazonaws.com:61617)"/>
</networkConnectors>
```

Netzwerk-Connectors für Broker4:

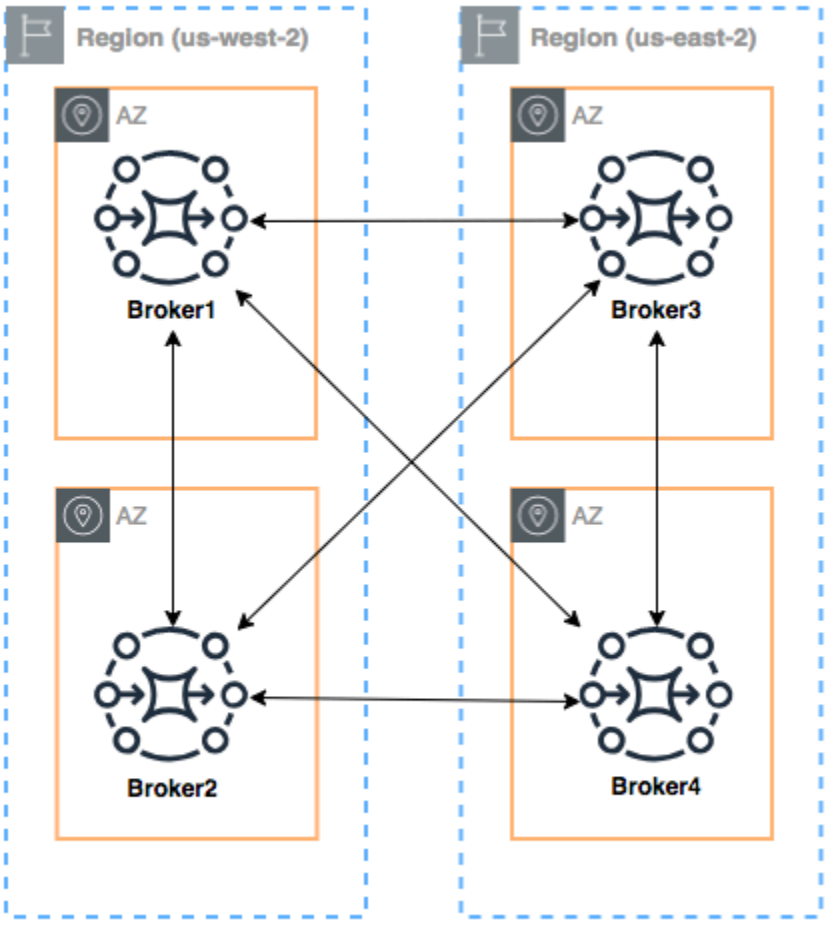
```
<networkConnectors>
    <networkConnector name="4_to_1" userName="myCommonUser" duplex="false"
        uri="static:(ssl://b-1234a5b6-78cd-901e-2fgh-3i45j6k178l9-1.mq.us-
east-2.amazonaws.com:61617)"/>
    <networkConnector name="4_to_2" userName="myCommonUser" duplex="false"
        uri="static:(ssl://b-1234a5b6-78cd-901e-2fgh-3i45j6k178l9-1.mq.us-
east-2.amazonaws.com:61617)"/>
</networkConnectors>
```

Netzwerk-Connectors für Broker5:

```
<networkConnectors>
    <networkConnector name="5_to_1" userName="myCommonUser" duplex="false"
        uri="static:(ssl://b-1234a5b6-78cd-901e-2fgh-3i45j6k178l9-1.mq.us-
east-2.amazonaws.com:61617)"/>
    <networkConnector name="5_to_2" userName="myCommonUser" duplex="false"
        uri="static:(ssl://b-9876l5k4-32ji-109h-8gfe-7d65c4b132a1-2.mq.us-
east-2.amazonaws.com:61617)"/>
</networkConnectors>
```

Regionsübergreifend

Um ein regionsübergreifendes AWS Brokernetzwerk zu konfigurieren, stellen Sie Broker in diesen Regionen bereit und konfigurieren Sie Netzwerkkonnektoren für die Endpunkte dieser Broker.



Um ein Netzwerk von Brokern wie in diesem Beispiel zu konfigurieren, können Sie `networkConnectors`-Einträge zu den Konfigurationen von Broker1 und Broker4 hinzufügen, die auf die Wire-Level-Endpunkte dieser Broker verweisen.

Netzwerk-Connectors für Broker1:

```
<networkConnectors>
    <networkConnector name="1_to_2" userName="myCommonUser" duplex="true"
        uri="static:(ssl://b-9876l5k4-32ji-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>
```

Netzwerk-Connector für 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>
```

Netzwerk-Connectors für Broker4:

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

Dynamisches Failover mit Transport Connectors

Zusätzlich zur Konfiguration von `networkConnector`-Elementen können Sie die `transportConnector`-Optionen Ihres Brokers zur Aktivierung von dynamischem Failover

konfigurieren und zum Neuausgleich der Verbindungen, wenn Broker dem Netzwerk hinzugefügt oder daraus entfernt werden.

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

In diesem Beispiel sind `updateClusterClients` und `rebalanceClusterClients` auf `true` gesetzt. In diesem Fall wird den Clients eine Liste von Brokern im Netzwerk präsentiert, und die Clients werden zur Neuausrichtung aufgefordert, wenn ein neuer Broker hinzukommt.

Verfügbare Optionen:

- `updateClusterClients`: Übergibt Clients Informationen zu Änderungen im Netzwerk der Brokertopologie.
- `rebalanceClusterClients` Lässt Clients eine Neuausrichtung über die Broker hinweg durchführen, wenn einem Brokernetzwerk ein neuer Broker hinzugefügt wird.
- `updateClusterClientsOnRemove`: Aktualisiert Clients mit Topologieinformationen, wenn ein Broker ein Brokernetzwerk verlässt.

Wenn `updateClusterClients` auf „true“ gesetzt ist, können Clients zur Verbindung mit einem einzelnen Broker in einem Brokernetzwerk konfiguriert werden.

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

Wenn ein neuer Broker eine Verbindung herstellt, erhält er eine Liste URIs aller Broker im Netzwerk. Wenn die Verbindung zu dem Broker fehlschlägt, kann er dynamisch zu einem anderen Broker wechseln, der bei seiner Verbindung verfügbar war.

Weitere Informationen zum Failover finden Sie unter Brokerseitige Failover-Optionen in der Active MQ-Dokumentation.

Konfigurationen für Amazon MQ für ActiveMQ Broker

Eine Konfiguration enthält alle Einstellungen für Ihren ActiveMQ-Broker im XML Format (ähnlich der ActiveMQ-Datei). `activemq.xml` Sie können eine Konfiguration erstellen, bevor Sie Broker erstellen. Sie können die Konfiguration dann auf einen oder mehrere Broker anwenden.

> **Important**
>
> Das Vornehmen von Änderungen an einer Konfigurationnichtwenden Sie die Änderungen sofort an den Broker an. Um Ihre Änderungen zu übernehmen, müssen Sie auf den nächsten Wartungszeitraum warten oder den Broker neu starten.
> Derzeit ist es nicht möglich, eine Konfiguration zu löschen.

Attribute

Eine Broker-Konfiguration verfügt über mehrere Attribute, z. B.:

- Einen Namen (`MyConfiguration`)
- Eine ID (`c-1234a5b6-78cd-901e-2fgh-3i45j6k178l9`)
- Ein Amazon-Ressourcenname (ARN) (`arn:aws:mq:us-east-2:123456789012:configuration:c-1234a5b6-78cd-901e-2fgh-3i45j6k178l9`)

Eine vollständige Liste der Konfigurationsattribute finden Sie im Folgenden in der Amazon MQ REST API MQ-Referenz:

- RESTVorgangs-ID: Konfiguration
- RESTVorgangs-ID: Konfigurationen

Eine vollständige Liste der Konfigurationsrevisions-Attribute finden Sie im folgenden Abschnitt:

- RESTVorgangs-ID: Revision der Konfiguration
- RESTVorgangs-ID: Konfigurationsrevisionen

Verwenden von XML Spring-Konfigurationsdateien

ActiveMQ-Broker werden mithilfe von XMLSpring-Dateien konfiguriert. Sie können viele Aspekte Ihres ActiveMQ-Brokers konfigurieren, wie z. B. vordefinierte Ziele, Ziel-Richtlinien, Autorisierungsrichtlinien und Plugins. Amazon MQ kontrolliert einige dieser Konfigurationselemente, wie z. B. Netzwerktransporte und Speicherung. Andere Konfigurationsoptionen, wie z. B. das Erstellen von Broker-Netzwerken, werden derzeit nicht unterstützt.

Der vollständige Satz der unterstützten Konfigurationsoptionen ist in den Amazon MQ XML MQ-Schemas angegeben. Laden Sie ZIP-Dateien der unterstützten Schemas unter Verwendung der folgenden Links herunter.

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

Sie können Ihre Konfigurationsdateien anhand dieses Schemas validieren und bereinigen. Mit Amazon MQ können Sie auch Konfigurationen bereitstellen, indem Sie Dateien hochladenXML. Wenn Sie eine XML Datei hochladen, bereinigt und entfernt Amazon MQ automatisch ungültige und verbotene Konfigurationsparameter gemäß dem Schema.

> **Note**
>
> Für Attribute sind nur statische Werte zulässig. Amazon MQ löscht Elemente und Attribute, die Spring-Ausdrücke, -Variablen und -Referenzen aus Ihrer Konfiguration enthalten.

Brokerkonfiguration für Amazon MQ für ActiveMQ erstellen

Eine Konfiguration enthält alle Einstellungen für Ihren ActiveMQ-Broker im XML Format (ähnlich der ActiveMQ-Datei). `activemq.xml` Sie können eine Konfiguration erstellen, bevor Sie Broker erstellen. Sie können die Konfiguration dann auf einen oder mehrere Broker anwenden. Sie können eine Konfiguration unmittelbar oder während eines Wartungsfensters übernehmen.

Weitere Informationen finden Sie hier:

- Amazon MQ Broker Configuration Parameters

Das folgende Beispiel zeigt, wie Sie eine Amazon MQ-Broker-Konfiguration mithilfe der AWS Management Console erstellen und anwenden.

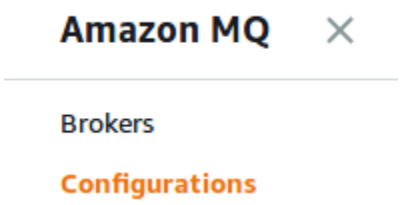
Themen

- Eine neue Konfiguration erstellen
- Erstellen einer neuen Konfigurationsversion
- Eine Konfigurationsrevision auf Ihren Broker anwenden

Eine neue Konfiguration erstellen

Um eine neue Broker-Konfiguration zu erstellen, erstellen Sie zunächst die neue Konfiguration.

1. Melden Sie sich bei der Amazon MQ-Konsole an.
2. Erweitern Sie den Navigationsbereich auf der linken Seite, und wählen Sie Configurations (Konfigurationen) aus.



3. Wählen Sie auf der Seite Configurations (Konfigurationen) die Option Create configuration (Konfiguration erstellen).
4. Geben Sie auf der Seite Create configuration (Konfiguration erstellen) im Abschnitt Details den Configuration name (Konfigurationsname) (z. B. `MyConfiguration`) ein und wählen Sie eine Broker-Engine-Version aus.

> **Note**
>
> Weitere Informationen zu ActiveMQ-Engine-Versionen, die von Amazon MQ für ActiveMQ unterstützt werden, finden Sie unterthe section called “Versionsverwaltung.”.

5. Wählen Sie Create configuration (Konfiguration erstellen).

Erstellen einer neuen Konfigurationsversion

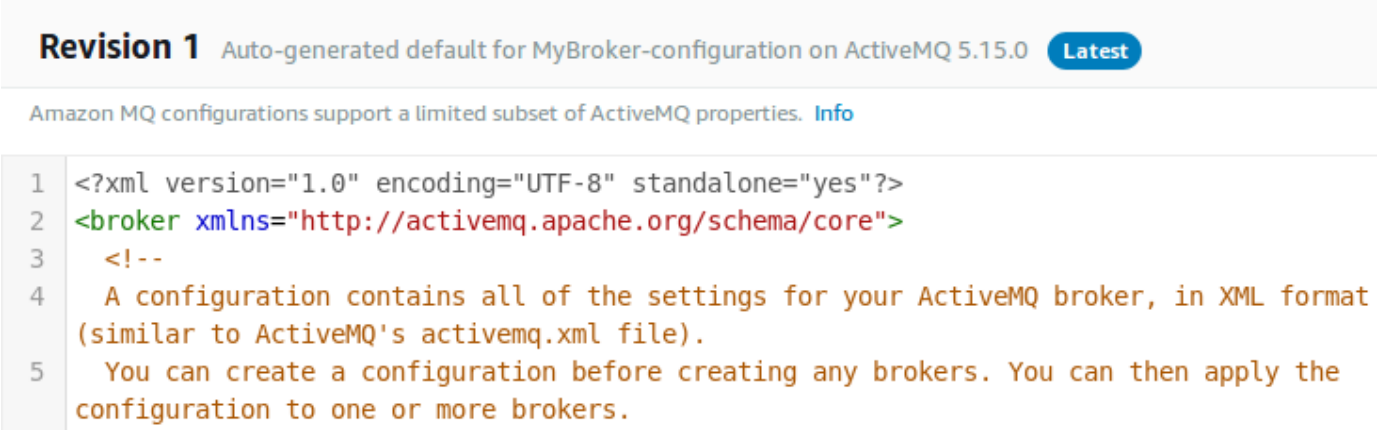
Nachdem Sie eine Broker-Konfiguration erstellt haben, müssen Sie die Konfiguration mithilfe einer Konfigurationsrevision bearbeiten.

1. Wählen Sie aus der Konfigurationsliste ***MyConfiguration***.

   > **Note**
   >
   > Die erste Revision der Konfiguration wird stets bei der Konfigurationserstellung durch Amazon MQ für Sie erstellt.

   Auf dem ***MyConfiguration***Seite, der Broker-Engine-Typ und die Version, die Ihre neue Konfigurationsrevision verwendet (z. B. Apache ActiveMQ 5.15.16), werden angezeigt.
2. Auf der Registerkarte „Konfigurationsdetails“ werden die Versionsnummer, die Beschreibung und das Format der Broker-Konfiguration angezeigt. XML

   > **Note**
   >
   > Durch die Bearbeitung der aktuellen Konfiguration wird eine neue Konfigurationsversion erstellt.

3. Wählen Sie Konfiguration bearbeiten aus und nehmen Sie Änderungen an der XML Konfiguration vor.
4. Wählen Sie Save (Speichern) aus.

DieSpeichern der Revisionwird angezeigt.

5. (Optional) Geben Sie `A description of the changes in this revision` ein.
6. Wählen Sie Save (Speichern) aus.

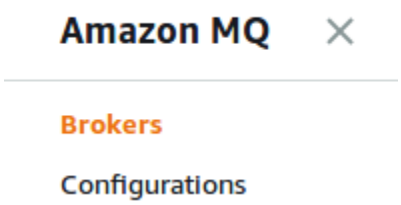
   Die neue Version der Konfiguration wird gespeichert.

> **Important**
>
> Die Amazon MQ Konsole löscht ungültige und nicht zulässige Konfigurationsparameter automatisch entsprechend eines Schemas. Weitere Informationen und eine vollständige Liste der zulässigen XML Parameter finden Sie unterAmazon MQ Broker Configuration Parameters.

Eine Konfigurationsrevision auf Ihren Broker anwenden

Nachdem Sie die Konfiguration überarbeitet haben, können Sie die Konfigurationsrevision auf Ihren Broker anwenden.

1. Erweitern Sie den Navigationsbereich auf der linken Seite, und wählen Sie Broker aus.



2. Wählen Sie in der Brokerliste Ihren Broker aus (z. B. MyBroker) und klicken Sie dann auf Bearbeiten.
3. Auf der Seite Bearbeiten ***`MyBroker`***Wählen Sie auf der Seite im Abschnitt Konfiguration eine Konfiguration und eine Revision aus und klicken Sie dann auf Änderungen planen.
4. Wählen Sie im Abschnitt Schedule broker modifications (Broker-Änderungen planen) aus, ob die Änderungen During the next scheduled maintenance window (Im nächsten geplanten Wartungsfenster) oder Immediately (Sofort) angewendet werden sollen.

> **Important**
>
> Ihr Broker ist offline, während er neu gestartet wird.

5. Wählen Sie Apply (Anwenden) aus.

   Ihre Konfigurationsversion wird zu der angegebenen Zeit auf Ihren Broker angewendet.

Bearbeiten Sie eine Konfigurationsrevision von Amazon MQ für ActiveMQ

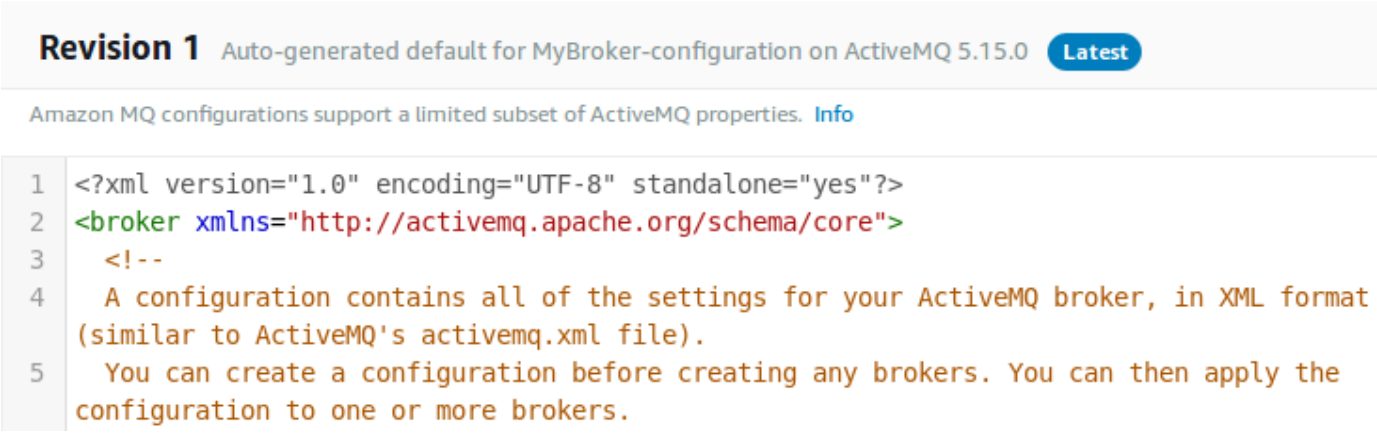
Möglicherweise möchten Sie eine Konfigurationsrevision bearbeiten, nachdem Sie sie auf Ihren Broker angewendet haben. Verwenden Sie die folgenden Anweisungen, um eine Konfigurationsrevision zu bearbeiten.

1. Melden Sie sich bei der Amazon-MQ-Konsole an.
2. Wählen Sie in der Brokerliste Ihren Broker aus (z. B. MyBroker) und klicken Sie dann auf Bearbeiten.
3. Auf der ***MyBroker***Wählen Sie auf der Seite Bearbeiten aus.
4. Auf der Seite Bearbeiten ***MyBroker***Wählen Sie auf der Seite im Abschnitt Konfiguration eine Konfiguration und eine Revision aus und klicken Sie dann auf Bearbeiten.

   > **Note**
   >
   > Wenn Sie beim Erstellen eines Brokers eine Konfiguration auswählen, wird die erste Revision der Konfiguration stets bei der Konfigurationserstellung durch Amazon MQ für Sie erstellt.

   Auf der ***MyBroker***Seite, der Broker-Engine-Typ und die Version, die die Konfiguration verwendet (z. B. Apache ActiveMQ 5.15.8), werden angezeigt.
5. Auf der Registerkarte „Konfigurationsdetails“ werden die Versionsnummer, die Beschreibung und das Format der Broker-Konfiguration angezeigt. XML

> **Note**
>
> Durch die Bearbeitung der aktuellen Konfiguration wird eine neue Konfigurationsversion erstellt.



6. Wählen Sie Konfiguration bearbeiten aus und nehmen Sie Änderungen an der XML Konfiguration vor.
7. Wählen Sie Save (Speichern) aus.

   DieSpeichern der Revisionwird angezeigt.
8. (Optional) Geben Sie `A description of the changes in this revision` ein.
9. Wählen Sie Save (Speichern) aus.

   Die neue Version der Konfiguration wird gespeichert.

> **Important**
>
> Die Amazon MQ Konsole löscht ungültige und nicht zulässige Konfigurationsparameter automatisch entsprechend eines Schemas. Weitere Informationen und eine vollständige Liste der zulässigen XML Parameter finden Sie unterAmazon MQ Broker Configuration Parameters.

In Amazon MQ MQ-Konfigurationen zulässige Elemente

Es folgt eine detaillierte Auflistung der in Amazon MQ-Konfigurationen zulässigen Elemente. Weitere Informationen finden Sie unter XMLKonfiguration in der Apache ActiveMQ-Dokumentation.

Element
`abortSlowAckConsumerStrategy` (Attribute)
`abortSlowConsumerStrategy` (Attribute)
`authorizationEntry` (Attribute)
`authorizationMap` (untergeordnete Sammlungselemente)
`authorizationPlugin` (untergeordnete Sammlungselemente)
`broker` (Attribute | untergeordnete Sammlungselemente)
`cachedMessageGroupMapFactory` (Attribute)
`compositeQueue` (Attribute | untergeordnete Sammlungselemente)
`compositeTopic` (Attribute | untergeordnete Sammlungselemente)
`constantPendingMessageLimitStrategy` (Attribute)
`discarding` (Attribute)
`discardingDLQBrokerPlugin` (Attribute)
`fileCursor`
`fileDurableSubscriberCursor`
`fileQueueCursor`
`filteredDestination` (Attribute)
`fixedCountSubscriptionRecoveryPolicy` (Attribute)

Element
`fixedSizedSubscriptionRecoveryPolicy` (Attribute)
`forcePersistencyModeBrokerPlugin` (Attribute)
`individualDeadLetterStrategy` (Attribute)
`lastImageSubscriptionRecoveryPolicy`
`messageGroupHashBucketFactory` (Attribute)
`mirroredQueue` (Attribute)
`noSubscriptionRecoveryPolicy`
`oldestMessageEvictionStrategy` (Attribute)
`oldestMessageWithLowestPriorityEvictionStrategy` (Attribute)
`policyEntry` (Attribute | untergeordnete Sammlungselemente)
`policyMap` (untergeordnete Sammlungselemente)
`prefetchRatePendingMessageLimitStrategy` (Attribute)
`priorityDispatchPolicy`
`priorityNetworkDispatchPolicy`
`queryBasedSubscriptionRecoveryPolicy` (Attribute)
`queue` (Attribute)
`redeliveryPlugin` (Attribute | untergeordnete Sammlungselemente)
`redeliveryPolicy` (Attribute)
`redeliveryPolicyMap` (untergeordnete Sammlungselemente)
`retainedMessageSubscriptionRecoveryPolicy` (untergeordnete Sammlungselemente)

Element
roundRobinDispatchPolicy
sharedDeadLetterStrategy (Attribute | untergeordnete Sammlungselemente)
simpleDispatchPolicy
simpleMessageGroupMapFactory
statisticsBrokerPlugin
storeCursor
storeDurableSubscriberCursor (Attribute)
strictOrderDispatchPolicy
tempDestinationAuthorizationEntry (Attribute)
tempQueue (Attribute)
tempTopic (Attribute)
timedSubscriptionRecoveryPolicy (Attribute)
timeStampingBrokerPlugin (Attribute)
topic (Attribute)
transportConnector (Attribute)
uniquePropertyMessageEvictionStrategy (Attribute)
virtualDestinationInterceptor (untergeordnete Sammlungselemente)
virtualTopic (Attribute)
vmCursor
vmDurableCursor

Element
vmQueueCursor

In Amazon MQ-Konfigurationen zulässige Elemente und ihre Attribute

Es folgt eine detaillierte Auflistung der in Amazon MQ-Konfigurationen zulässigen Elemente und deren Attribute. Weitere Informationen finden Sie unter XMLKonfiguration in der Apache ActiveMQ-Dokumentation.

Element	Attribut
abortSlowAckConsumerStrategy	abortConnection
	checkPeriod
	ignoreIdleConsumers
	ignoreNetworkConsumers
	maxSlowCount
	maxSlowDuration
	maxTimeSinceLastAck
	name
abortSlowConsumerStrategy	abortConnection
	checkPeriod
	ignoreNetworkConsumers
	maxSlowCount
	maxSlowDuration
	name

Element	Attribut
authorizationEntry	admin
	queue
	read
	tempQueue
	tempTopic
	topic
	write
broker	advisorySupport
	allowTempAutoCreationOnSend
	cacheTempDestinations
	consumerSystemUsagePortion
	dedicatedTaskRunner
	deleteAllMessagesOnStartup
	keepDurableSubsActive
	enableMessageExpirationOnActiveDurableSubs
	maxPurgedDestinationsPerSweep
	maxSchedulerRepeatAllowed
	monitorConnectionSplits
	networkConnectorStartAsync

Element	Attribut
	offlineDurableSubscriberTaskSchedule
	offlineDurableSubscriberTimeout
	persistenceThreadPriority
	persistent
	populateJMSXUserID
	producerSystemUsagePortion
	rejectDurableConsumers
	rollbackOnlyOnAsyncException
	schedulePeriodForDestinationPurge
	schedulerSupport
	splitSystemUsageForProducersConsumers
	taskRunnerPriority
	timeBeforePurgeTempDestinations
	useAuthenticatedPrincipalForJMSXUserID
	useMirroredQueues
	useTempMirroredQueues
	useVirtualDestSubs
	useVirtualDestSubsOnCreation

Element	Attribut
	`useVirtualTopics`
`cachedMessageGroupMapFactory`	`cacheSize`
`compositeQueue`	`concurrentSend`
	`copyMessage`
	`forwardOnly`
	`name`
	`sendWhenNotMatched`
`compositeTopic`	`concurrentSend`
	`copyMessage`
	`forwardOnly`
	`name`
	`sendWhenNotMatched`
conditionalNetworkBridgeFilterFactory	`rateDuration`
	`rateLimit`
	`replayDelay`
	`replayWhenNoConsumers`
	`selectorAware` Unterstützt in Apache ActiveMQ 5.16.x

Element	Attribut
constantPendingMessageLimitStrategy	limit
discarding	deadLetterQueue
	enableAudit
	expiration
	maxAuditDepth
	maxProducersToAudit
	processExpired
	processNonPersistent
discardingDLQBrokerPlugin	dropAll
	dropOnly
	dropTemporaryQueues
	dropTemporaryTopics
	reportInterval
filteredDestination	queue
	selector
	topic
fixedCountSubscriptionRecoveryPolicy	maximumSize
fixedSizedSubscriptionRecoveryPolicy	maximumSize
	useSharedBuffer

Element	Attribut
forcePersistencyModeBrokerPlugin	persistenceFlag
individualDeadLetterStrategy	destinationPerDurableSubscriber
	enableAudit
	expiration
	maxAuditDepth
	maxProducersToAudit
	processExpired
	processNonPersistent
	queuePrefix
	queueSuffix
	topicPrefix
	topicSuffix
	useQueueForQueueMessages
	useQueueForTopicMessages
messageGroupHashBucketFactory	bucketCount
	cacheSize
mirroredQueue	copyMessage
	postfix
	prefix
oldestMessageEvictionStrategy	evictExpiredMessagesHighWatermark

Element	Attribut
oldestMessageWithLowestPrio rityEvictionStrategy	evictExpiredMessagesHighWat ermark
policyEntry	advisoryForConsumed
	advisoryForDelivery
	advisoryForDiscardingMessages
	advisoryForFastProducers
	advisoryForSlowConsumers
	advisoryWhenFull
	allConsumersExclusiveByDefault
	alwaysRetroactive
	blockedProducerWarningInterval
	consumersBeforeDispatchStarts
	cursorMemoryHighWaterMark
	doOptimzeMessageStorage
	durableTopicPrefetch
	enableAudit
	expireMessagesPeriod
	gcInactiveDestinations
	gcWithNetworkConsumers
	inactiveTimeoutBeforeGC
	inactiveTimoutBeforeGC

Element	Attribut
	includeBodyForAdvisory
	lazyDispatch
	maxAuditDepth
	maxBrowsePageSize
	maxDestinations
	maxExpirePageSize
	maxPageSize
	maxProducersToAudit
	maxQueueAuditDepth
	memoryLimit
	messageGroupMapFactoryType
	minimumMessageSize
	optimizedDispatch
	optimizeMessageStoreInFligh tLimit
	persistJMSRedelivered
	prioritizedMessages
	producerFlowControl
	queue
	queueBrowserPrefetch
	queuePrefetch

Element	Attribut
	`reduceMemoryFootprint`
	`sendAdvisoryIfNoConsumers`
	`sendFailIfNoSpace`
	`sendFailIfNoSpaceAfterTimeout` Unterstützt in Apache ActiveMQ 5.16.4 und höher
	`sendDuplicateFromStoreToDLQ`
	`storeUsageHighWaterMark`
	`strictOrderDispatch`
	`tempQueue`
	`tempTopic`
	`timeBeforeDispatchStarts`
	`topic`
	`topicPrefetch`
	`useCache`
	`useConsumerPriority`
`usePrefetchExtension`	
`prefetchRatePendingMessageLimitStrategy`	`multiplier`

Element	Attribut
queryBasedSubscriptionRecoveryPolicy	query
queue	DLQ
	physicalName
redeliveryPlugin	fallbackToDeadLetter
	sendToDlqIfMaxRetriesExceeded
redeliveryPolicy	backOffMultiplier
	collisionAvoidancePercent
	initialRedeliveryDelay
	maximumRedeliveries
	maximumRedeliveryDelay
	preDispatchCheck
	queue
	redeliveryDelay
	tempQueue
	tempTopic
	topic
	useCollisionAvoidance
	useExponentialBackOff
sharedDeadLetterStrategy	enableAudit
	expiration

Element	Attribut
	maxAuditDepth
	maxProducersToAudit
	processExpired
	processNonPersistent
storeDurableSubscriberCursor	immediatePriorityDispatch
	useCache
tempDestinationAuthorizationEntry	admin
	queue
	read
	tempQueue
	tempTopic
	topic
	write
tempQueue	DLQ
	physicalName
tempTopic	DLQ
	physicalName
timedSubscriptionRecoveryPolicy	zeroExpirationOverride
timeStampingBrokerPlugin	recoverDuration
	futureOnly

Element	Attribut
	processNetworkMessages
	ttlCeiling
topic	DLQ
	physicalName
transportConnector	•
	name
	updateClusterClients
	rebalanceClusterClients
	updateClusterClientsOnRemove
uniquePropertyMessageEvictionStrategy	evictExpiredMessagesHighWatermark
	propertyName
virtualTopic	concurrentSend
	local
	dropOnResourceLimit
	name
	postfix
	prefix
	selectorAware
	setOriginalDestination

Element	Attribut
	`transactedSend`

Attribute des übergeordneten Amazon MQ Elemente

Im Folgenden finden Sie eine detaillierte Erklärung der Attribute des übergeordneten Elements. Weitere Informationen finden Sie unter XMLKonfiguration in der Apache ActiveMQ-Dokumentation.

Themen

- broker

broker

`broker` ist ein übergeordnetes Sammlungselement.

Attribute

networkConnectionStartAsynchron

Um die Netzwerklatenz zu minimieren und anderen Netzwerken einen rechtzeitigen Start zu ermöglichen, verwenden Sie das Tag `<networkConnectionStartAsync>`. Das Tag weist den Broker an, über einen Executor Netzwerkverbindungen parallel und asynchron zu einem Brokerstart zu starten.

Standardwert: `false`

Beispielkonfiguration

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

In Amazon MQ-Konfigurationen zulässige Elemente, untergeordnete Sammlungselemente und deren untergeordnete Attribute

Es folgt eine detaillierte Auflistung der in Holen Sie sich die Amazon MQ-Konfigurationen zulässigen Elemente, untergeordneten Sammlungselemente und deren untergeordneten Attribute. Weitere Informationen finden Sie unter XMLKonfiguration in der Apache ActiveMQ-Dokumentation.

Element	Untergeordnetes Sammlungselement	Untergeordnetes Element
authorizationMap	authorizationEntries	authorizationEntry
		tempDestinationAuthorizationEntry
	defaultEntry	authorizationEntry
		tempDestinationAuthorizationEntry
	tempDestinationAuthorizationEntry	tempDestinationAuthorizationEntry
authorizationPlugin	map	authorizationMap
broker	destinationInterceptors	mirroredQueue
		virtualDestinationInterceptor
	destinationPolicy	policyMap
	destinations	queue
		tempQueue
		tempTopic
		topic
	networkConnectors	networkConnector
	persistenceAdapter	kahaDB
	plugins	authorizationPlugin
		discardingDLQBrokerPlugin

Element	Untergeordnetes Sammlungselement	Untergeordnetes Element
		forcePersistencyModeBrokerPlugin
		redeliveryPlugin
		statisticsBrokerPlugin
		timeStampingBrokerPlugin
	systemUsage	systemUsage
	transportConnector	name
		updateClusterClients
		rebalanceClusterClients
		updateClusterClientsOnRemove
compositeQueue	forwardTo	queue
		tempQueue
		tempTopic
		topic
		filteredDestination
compositeTopic	forwardTo	queue
		tempQueue
		tempTopic

Element	Untergeordnetes Sammlungselement	Untergeordnetes Element
		topic
		filteredDestination
policyEntry	deadLetterStrategy	discarding
		individualDeadLetterStrategy
		sharedDeadLetterStrategy
	destination	queue
		tempQueue
		tempTopic
		topic
	dispatchPolicy	priorityDispatchPolicy
		priorityNetworkDispatchPolicy
		roundRobinDispatchPolicy
		simpleDispatchPolicy
		strictOrderDispatchPolicy
		clientIdFilterDispatchPolicy

Element	Untergeordnetes Sammlungselement	Untergeordnetes Element
	messageEvictionStrategy	oldestMessageEvictionStrategy
		oldestMessageWithLowestPriorityEvictionStrategy
		uniquePropertyMessageEvictionStrategy
	messageGroupMapFactory	cachedMessageGroupMapFactory
		messageGroupHashBucketFactory
		simpleMessageGroupMapFactory
	pendingDurableSubscriberPolicy	fileDurableSubscriberCursor
		storeDurableSubscriberCursor
		vmDurableCursor
	pendingMessageLimitStrategy	constantPendingMessageLimitStrategy
		prefetchRatePendingMessageLimitStrategy
	pendingQueuePolicy	fileQueueCursor
		storeCursor

Element	Untergeordnetes Sammlungselement	Untergeordnetes Element
		vmQueueCursor
	pendingSubscriberPolicy	fileCursor
		vmCursor
	slowConsumerStrategy	abortSlowAckConsumerStrategy
		abortSlowConsumerStrategy
	subscriptionRecoveryPolicy	fixedCountSubscriptionRecoveryPolicy
		fixedSizedSubscriptionRecoveryPolicy
		lastImageSubscriptionRecoveryPolicy
		noSubscriptionRecoveryPolicy
		queryBasedSubscriptionRecoveryPolicy
		retainedMessageSubscriptionRecoveryPolicy
timedSubscriptionRecoveryPolicy		
policyMap	defaultEntry	policyEntry
	policyEntries	policyEntry

Element	Untergeordnetes Sammlungselement	Untergeordnetes Element
redeliveryPlugin	redeliveryPolicyMap	redeliveryPolicyMap
redeliveryPolicyMap	defaultEntry	redeliveryPolicy
	redeliveryPolicyEntries	redeliveryPolicy
retainedMessageSubscriptionRecoveryPolicy	wrapped	fixedCountSubscriptionRecoveryPolicy
		fixedSizedSubscriptionRecoveryPolicy
		lastImageSubscriptionRecoveryPolicy
		noSubscriptionRecoveryPolicy
		queryBasedSubscriptionRecoveryPolicy
		retainedMessageSubscriptionRecoveryPolicy
		timedSubscriptionRecoveryPolicy
sharedDeadLetterStrategy	deadLetterQueue	queue
		tempQueue
		tempTopic
		topic

Element	Untergeordnetes Sammlungselement	Untergeordnetes Element
`virtualDestinationInterceptor`	`virtualDestinations`	`compositeQueue`
		`compositeTopic`
		`virtualTopic`

Amazon MQ-Attribute

Im Folgenden finden Sie eine detaillierte Erklärung der Attribute untergeordneter Sammlungselemente. Weitere Informationen finden Sie unter XMLKonfiguration in der Apache ActiveMQ-Dokumentation.

Themen

- authorizationEntry
- networkConnector
- kahaDB
- systemUsage

authorizationEntry

`authorizationEntry` ist ein untergeordnetes Attribut des untergeordneten Sammlungselements `authorizationEntries`.

Attribute

admin|read|write

Die Berechtigungen, die einer Gruppe von Benutzern gewährt werden. Weitere Informationen finden Sie unter Immer eine Autorisierungszuordnung konfigurieren.

Wenn Sie eine Autorisierungszuweisung angeben, die die`activemq-webconsole`können Sie die ActiveMQ Webkonsole nicht verwenden, da die Gruppe nicht berechtigt ist, Nachrichten an den Amazon MQ -Broker zu senden oder von ihm Nachrichten zu empfangen.

Standardwert: `null`

Beispielkonfiguration

```
<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` ist ein untergeordnetes Attribut des untergeordneten Sammlungselements `networkConnectors`.

Themen

- Attribute
- Beispielkonfigurationen

Attribute

conduitSubscriptions

Gibt an, ob eine Netzwerkverbindung in einem Netzwerk von Brokern mehrere Verbraucher, die am gleichen Ziel angemeldet sind, als einzelnen Verbraucher behandelt. Beispiel: Wenn `conduitSubscriptions` auf `true` gestellt ist und zwei Verbraucher mit dem Broker B verbunden sind und von einem Ziel aus konsumieren, kombiniert der Broker B die Abonnements zu einem einzigen logischen Abonnement über die Netzwerkverbindung zum Broker A, sodass nur eine einzige Kopie einer Nachricht vom Broker A an den Broker B weitergeleitet wird.

> **Note**
>
> Durch Festlegen von `conduitSubscriptions` auf `true` können Sie den redundanten Netzwerkverkehr reduzieren. Die Verwendung dieses Attributs kann jedoch Auswirkungen auf den Lastenausgleich von Nachrichten zwischen Verbrauchern haben und in bestimmten Szenarien zu falschem Verhalten führen (z. B. bei JMS Nachrichtenselektoren oder dauerhaften Themen).

Standardwert: `true`

duplex

Gibt an, ob die Verbindung im Netzwerk der Broker verwendet wird, um Nachrichten zu produzieren und zu konsumieren. Wenn beispielsweise der Broker A eine Verbindung zum Broker B im Nicht-Duplex-Modus herstellt, können Nachrichten nur vom Broker A an den Broker B weitergeleitet werden. Wenn der Broker A jedoch eine Duplexverbindung zum Broker B herstellt, kann der Broker B Nachrichten an den Broker A weiterleiten, ohne einen `<networkConnector>`.

Standardwert: `false`

Name

Der Name der Brücke im Netzwerk von Brokern.

Standardwert: `bridge`

uri

Der Wire-Level-Protokoll-Endpunkt für einen von zwei Brokern (oder für mehrere Broker) in einem Netzwerk von Brokern.

Standardwert: `null`

username

Der Benutzername, der den Brokern in einem Netzwerk von Brokern gemeinsam ist.

Standardwert: `null`

Beispielkonfigurationen

> **Note**
>
> Bei der Verwendung eines `networkConnector` zur Definition eines Netzwerk von Brokern geben Sie das Passwort für den gemeinsamen Benutzer Ihrer Broker nicht an.

Ein Netzwerk von Brokern mit zwei Brokern

In dieser Konfiguration sind zwei Broker in einem Netzwerk von Brokern verbunden. Der Name des Netzwerkconnectors ist`connector_1_to_2`, der gemeinsame Benutzername der Broker lautet, die Verbindung ist`myCommonUser`, und dem OpenWire Endpunkt URI wird ein Präfix vorangestellt `duplexstatic:`, was auf eine one-to-one Verbindung zwischen den Brokern hinweist.

```
<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>
```

Weitere Informationen finden Sie unter Configure Network Connectors for Your Broker.

Ein Netzwerk von Brokern mit mehreren Brokern

In dieser Konfiguration sind mehrere Broker in einem Netzwerk von Brokern verbunden. Der Name des Netzwerkconnectors ist`connector_1_to_2`, der gemeinsame Benutzername der Broker lautet, die Verbindung ist`myCommonUser`, und der kommagetrennten Liste der OpenWire Endpunkte URIs wird `duplex` ein Präfix vorangestellt`masterslave:`, was auf eine Failover-Verbindung zwischen den Brokern hinweist. Das Failover von Broker zu Broker ist nicht zufällig und Wiederherstellungsversuche dauern unbegrenzt an.

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

> **Note**
>
> Wir empfehlen die Verwendung des Präfixes `masterslave:` für Netzwerke von Brokern. Das Präfix ist identisch mit der expliziteren `static:failover:()?randomize=false&maxReconnectAttempts=0`-Syntax.

> **Note**
>
> Diese XML Konfiguration erlaubt keine Leerzeichen.

kahaDB

`kahaDB` ist ein untergeordnetes Attribut des untergeordneten Sammlungselements `persistenceAdapter`.

Attribute

concurrentStoreAndDispatchQueues

Gibt an, ob für Warteschlangen die gleichzeitige Speicherung und Verteilung verwendet werden soll. Weitere Informationen finden Sie unter Gleichzeitige Speicherung und Bereitstellung für Warteschlangen mit langsamen Konsumenten deaktivieren.

Standardwert: `true`

cleanupOnStop

> **Unterstützt in**
>
> Apache ActiveMQ 15.16.x und höher

Bei Deaktivierung erfolgt die Garbage Collection und Bereinigung nicht, wenn der Broker gestoppt wird, wodurch der Herunterfahrvorgang beschleunigt wird. Die erhöhte Geschwindigkeit ist in Fällen mit großen Datenbanken oder Scheduler-Datenbanken nützlich.

Standardwert: `true`

journalDiskSyncIntervall

Intervall (ms), wann eine Datenträgersynchronisierung durchgeführt werden soll, wenn `journalDiskSyncStrategy=periodic`. Weitere Informationen finden Sie in der Dokumentation zu Apache ActiveMQ KahaDB.

Standardwert: `1000`

journalDiskSyncStrategie

> Unterstützt in
>
> Apache ActiveMQ 15.14.x und höher

Konfiguriert die Richtlinie für die Datenträgersynchronisierung. Weitere Informationen finden Sie in der Dokumentation zu Apache ActiveMQ KahaDB.

Standardwert: `always`

> Note
>
> Laut der Dokumentation zu ActiveMQ ist der Datenverlust auf die Dauer von `journalDiskSyncInterval` begrenzt; der Standardwert beträgt 1 Sekunde. Der Datenverlust kann länger als das Intervall sein. Es ist jedoch schwierig, genaue Angaben zu machen. Gehen Sie vorsichtig vor.

preallocationStrategy

Konfiguriert, wie der Broker versucht, die Journaldateien vorab zuzuweisen, wenn eine neue Journaldatei benötigt wird. Weitere Informationen finden Sie in der Dokumentation zu Apache ActiveMQ KahaDB.

Standardwert: `sparse_file`

Beispielkonfiguration

Example

```
<broker xmlns="http://activemq.apache.org/schema/core">
```

```
                                <persistenceAdapter>
                                <kahaDB preallocationStrategy="zeros"
concurrentStoreAndDispatchQueues="false" journalDiskSyncInterval="10000"
journalDiskSyncStrategy="periodic"/>
                            </persistenceAdapter>
                            </broker>
```

systemUsage

`systemUsage` ist ein untergeordnetes Attribut des untergeordneten Sammlungselements `systemUsage`. Es steuert die maximale Menge an Speicherplatz, die der Broker verwendet, bevor die Produzenten verlangsamt werden. Weitere Informationen finden Sie unter Producer Flow Control in der Dokumentation zu Apache ActiveMQ.

Untergeordnetes Element

memoryUsage

`memoryUsage` ist ein untergeordnetes Element des untergeordneten Elements `systemUsage`. Es verwaltet die Speicherauslastung. Verwenden Sie `memoryUsage`, um nachzuverfolgen, wie viel von etwas verwendet wird, damit Sie die Nutzung von Arbeitssätzen produktiv steuern können. Weitere Informationen finden Sie im Schema in der Dokumentation zu Apache ActiveMQ.

Untergeordnetes Element

`memoryUsage` ist ein untergeordnetes Element des untergeordneten Elements `memoryUsage`.

Attribut

percentOfJvmHaufen

Ganzzahl zwischen 0 (inklusive) und 70 (inklusive).

Standardwert: `70`

Attribute

sendFailIfNoSpace

Legt fest, ob eine `send()`-Methode fehlschlagen soll, wenn kein freier Speicherplatz verfügbar ist. Der Standardwert lautet false, wodurch die `send()`-Methode so lange blockiert wird, bis

Speicherplatz verfügbar ist. Weitere Informationen finden Sie im Schema in der Dokumentation zu Apache ActiveMQ.

Standardwert: `false`

sendFailIfNoSpaceAfterTimeout

Standardwert: `null`

Beispielkonfiguration

Example

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

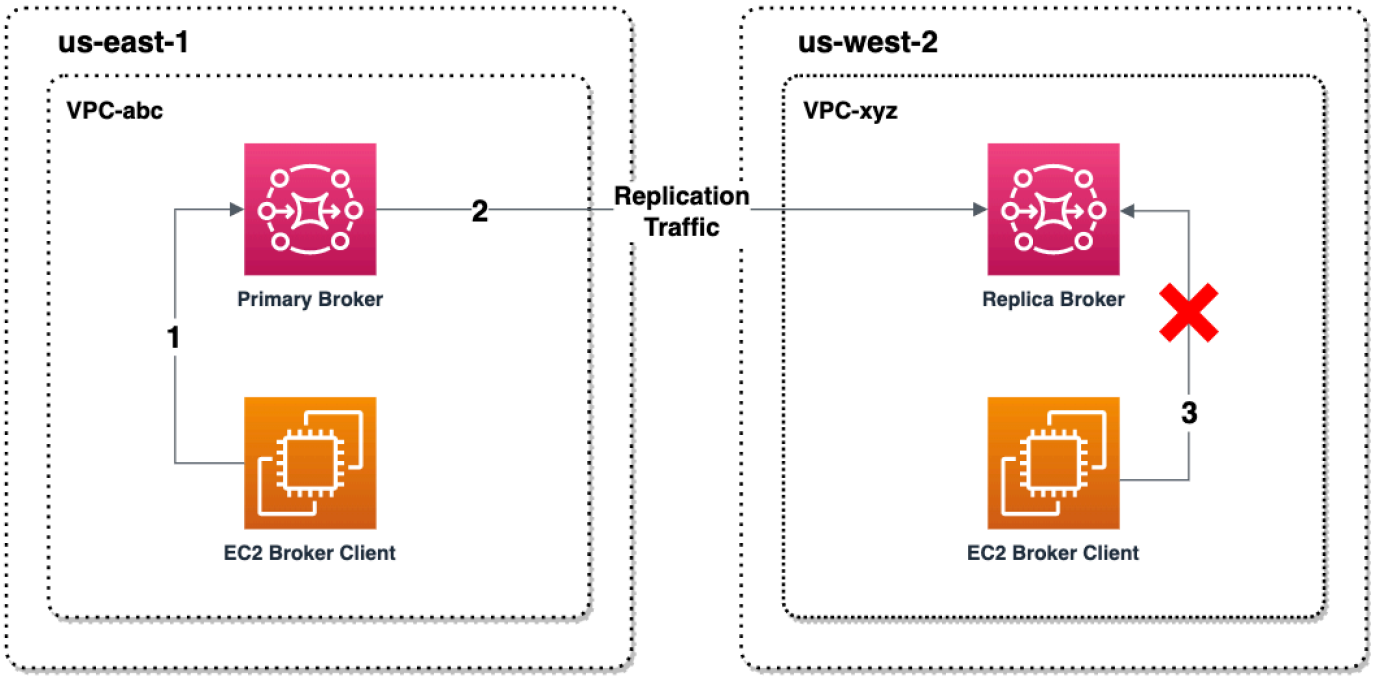
Regionsübergreifende Datenreplikation für Amazon MQ für ActiveMQ

Amazon MQ for ActiveMQ bietet eine Funktion zur regionsübergreifenden Datenreplikation (CRDR), die eine asynchrone Nachrichtenreplikation vom primären Broker in einer primären AWS Region zum Replikatbroker in einer Replikatregion ermöglicht. Durch das Senden einer Failover-Anfrage an den Amazon MQ API wird der aktuelle Replikatbroker zur Rolle des primären Brokers heraufgestuft, und der aktuelle primäre Broker wird zur Replikatrolle herabgestuft.

Primär- und Replikatbroker für die regionsübergreifende Datenreplikation

Sie können Primär- und Replikatbroker für die asynchrone Datenreplikation vom primären Broker in einer primären AWS Region zum Replikatbroker in einer Replikatregion erstellen. Die primäre Region besteht aus einem redundanten Paar aktiver/Standby-Broker, die als Primär-Broker bezeichnet werden. Die sekundäre Region besteht aus einem redundanten Paar aktiver/Standby-Broker, die als Replikat-Broker bezeichnet werden.

Das folgende Diagramm zeigt einen Replikat-Broker in einer sekundären Region, der asynchrone replizierte Daten vom Primär-Broker in der primären Region empfängt.



Primär- und Replikat-Broker fungieren als regionsübergreifende Datenwiederherstellungslösung. Wenn der Primär-Broker in der primären Region ausfällt, können Sie den Replikat-Broker in der sekundären Region zum Primär-Broker hochstufen, indem Sie ein Switchover oder Failover einleiten. Der ehemalige Primär-Broker wird dann zum Replikat-Broker und der ehemalige Replikat-Broker wird zum Primär-Broker hochgestuft. Anweisungen zum Erstellen eines Primär- und Replikat-Brokers finden Sie unter Einen Amazon MQ-Broker für die regionsübergreifende Datenreplikation erstellen.

> **Note**
>
> Nur für aktive/Standby-Broker verfügbar.

Einen Amazon MQ-Broker für die regionsübergreifende Datenreplikation erstellen

Mit der regionsübergreifenden Datenreplikation (CRDR) können Sie je nach Bedarf zwischen Amazon MQ for ActiveMQ-Nachrichtenmaklern in zwei AWS Regionen wechseln. Sie können einen vorhandenen Broker als Primär-Broker bestimmen und ein Replikat für diesen Broker erstellen oder

einen neuen Primär- sowie einen Replikat-Broker zusammen erstellen. Anschließend können Sie den Replikatbroker mithilfe des Amazon `Promote` API MQ-Vorgangs zur Rolle des primären Brokers heraufstufen. Weitere Informationen zu Primär- und Replikat-Brokern finden Sie unter Primär- und Replikatbroker für die regionsübergreifende Datenreplikation.

In der folgenden Anleitung wird beschrieben, wie Sie einen Replikat-Broker mithilfe der Amazon-MQ-Managementkonsole erstellen und konfigurieren können.

Themen

Voraussetzungen

Um das Feature für die regionsübergreifende Datenreplikation verwenden zu können, müssen Sie die folgenden Voraussetzungen überprüfen und erfüllen:

- Version: Das Feature für regionsübergreifende Datenreplikation ist nur für Broker von Amazon MQ für ActiveMQ in den Versionen 5.17.6 und höher verfügbar.
- Region: Die regionsübergreifende Datenreplikation wird in den folgenden Regionen unterstützt: USA Ost (Ohio), USA Ost (Nord-Virginia), USA West (Oregon) und USA West (Nordkalifornien).
- Instance-Typ: Die regionsübergreifende Datenreplikation ist nur für die Broker-Instance-Größen `mq.m5.large` und höher verfügbar.
- Bereitstellungstyp: Die regionsübergreifende Datenreplikation ist nur für Aktiv-/Standby-Broker mit einer Bereitstellung in mehreren Verfügbarkeitszonen verfügbar.
- Broker-Status: Sie können einen Replikat-Broker nur für einen primären Broker mit dem Broker-Status `Running` erstellen.

Schritt 1 (Optional): Erstellen eines neuen Primär-Brokers

Neuen Primär-Broker erstellen

1. Melden Sie sich bei der Amazon-MQ-Konsole an.
2. Wählen Sie auf der Seite „Brokers“ der Amazon-MQ-Konsole die Option Broker erstellen aus.

3. Wählen Sie auf der Seite Broker-Engine auswählen die Option Apache ActiveMQ aus.
4. Gehen Sie auf der Seite Auswählen von Bereitstellung und Speicher im Abschnitt Bereitstellungsmodus und Speichertyp folgendermaßen vor:

   - Wählen Sie den Bereitstellungsmodus aus (z. B. Aktiver/Standby-Broker). Ein aktiver/Standby-Broker besteht aus zwei Brokern in zwei verschiedenen Availability Zones, die in einem redundanten Paar konfiguriert sind. Diese Broker kommunizieren synchron mit Ihrer Anwendung und mit AmazonEFS. Weitere Informationen finden Sie unter Bereitstellungsoptionen für Amazon MQ für ActiveMQ-Broker.
5. Wählen Sie Weiter aus.
6. Gehen Sie auf der Seite Einstellungen konfigurieren im Abschnitt Details wie folgt vor:

   a. Geben Sie den Broker-Namen ein.

   > **Important**
   >
   > Fügen Sie den Namen der Makler keine personenbezogenen Daten (PII) oder andere vertrauliche oder sensible Informationen hinzu. Broker-Namen sind für andere AWS Dienste zugänglich, einschließlich CloudWatch Logs. Brokernamen sind nicht für private oder sensible Daten gedacht.

   b. Wählen Sie den Broker-Instance-Typ (z. B. mq.m5.large). Weitere Informationen finden Sie unter Broker instance types.
7. Geben Sie im Abschnitt Zugriff auf ActiveMQ-Webkonsole einen Benutzernamen und ein Passwort an. Die folgenden Einschränkungen gelten in Bezug auf Benutzernamen und Passwörter des Brokers:

   - Ihr Benutzername darf nur alphanumerische Zeichen, Bindestriche, Punkte, Unterstriche und Tilden (- . _ ~) enthalten.
   - Ihr Passwort muss mindestens 12 Zeichen lang sein, muss mindestens 4 eindeutige Zeichen enthalten und darf keine Kommas, Doppelpunkte oder Gleichheitszeichen (,:=) enthalten.

   > **Important**
   >
   > Fügen Sie den Broker-Benutzernamen keine personenbezogenen Daten (PII) oder andere vertrauliche oder sensible Informationen hinzu. Broker-Benutzernamen

sind für andere AWS Dienste zugänglich, einschließlich CloudWatch Logs. Broker-Benutzernamen sind nicht für private oder sensible Daten gedacht.

Der grüne, blinkende Balken oben auf der Seite bestätigt, dass Amazon MQ den Replikat-Broker in der Wiederherstellungsregion erstellt. Sie können auch die CRDR Rolle und den RPO Status Ihrer Makler einsehen. Um die Spalten CRDR Rolle und RPO Status zu deaktivieren, wählen Sie das Zahnradsymbol in der oberen rechten Ecke der Broker-Tabelle. Deaktivieren Sie dann auf der Seite „Einstellungen“ die Option „CRDRRolle“ oder RPO „Status“.

Schritt 2: Erstellen eines Replikats eines vorhandenen Brokers

1. Wählen Sie auf der Seite „Brokers“ der Amazon-MQ-Konsole die Option Replikat-Broker erstellen aus.
2. Wählen Sie auf der Seite „Primären Broker auswählen“ einen vorhandenen Broker aus, der als CRDR primärer Broker verwendet werden soll. Wählen Sie anschließend Weiter.
3. Wählen Sie auf der Seite Replikat-Broker konfigurieren im Dropdown-Menü die Replikationsregion aus.
4. Geben Sie im Abschnitt ActiveMQ-Konsolenbenutzer für Replikat-Broker einen Benutzernamen und ein Passwort für den Benutzer der Replikat-Broker-Konsole ein. Die folgenden Einschränkungen gelten in Bezug auf Benutzernamen und Passwörter des Brokers:

   - Ihr Benutzername darf nur alphanumerische Zeichen, Bindestriche, Punkte, Unterstriche und Tilden (- . _ ~) enthalten.
   - Ihr Passwort muss mindestens 12 Zeichen lang sein, muss mindestens 4 eindeutige Zeichen enthalten und darf keine Kommas, Doppelpunkte oder Gleichheitszeichen (,:=) enthalten.

   Important

   Fügen Sie den Broker-Benutzernamen keine personenbezogenen Daten (PII) oder andere vertrauliche oder sensible Informationen hinzu. Broker-Benutzernamen sind für andere AWS Dienste zugänglich, einschließlich CloudWatch Logs. Broker-Benutzernamen sind nicht für private oder sensible Daten gedacht.

5. Geben Sie im Abschnitt Datenreplikationsbenutzer zur Überbrückung des Zugriffs zwischen Brokern einen Benutzernamen und ein Passwort für den Benutzer ein, der sowohl auf den

Primär- als auch auf den Replikat-Broker zugreifen soll. Die folgenden Einschränkungen gelten in Bezug auf Benutzernamen und Passwörter des Brokers:

- Ihr Benutzername darf nur alphanumerische Zeichen, Bindestriche, Punkte, Unterstriche und Tilden (- . _ ~) enthalten.
- Ihr Passwort muss mindestens 12 Zeichen lang sein, muss mindestens 4 eindeutige Zeichen enthalten und darf keine Kommas, Doppelpunkte oder Gleichheitszeichen (,:=) enthalten.

> **Important**
>
> Fügen Sie den Broker-Benutzernamen keine personenbezogenen Daten (PII) oder andere vertrauliche oder sensible Informationen hinzu. Broker-Benutzernamen sind für andere AWS Dienste zugänglich, einschließlich CloudWatch Logs. Broker-Benutzernamen sind nicht für private oder sensible Daten gedacht.

Konfigurieren Sie alle zusätzlichen Einstellungen. Wählen Sie anschließend Weiter aus.

6. Prüfen Sie auf der Seite Überprüfen und erstellen die Details des Replikat-Brokers. Wählen Sie dann Replikat-Broker erstellen aus.
7. Starten Sie anschließend den Primär-Broker neu. Dadurch wird auch der Replikat-Broker neu gestartet. Anleitungen zum Neustart Ihres Brokers finden Sie unter Rebooting a Broker.

Weitere Informationen zur Konfiguration zusätzlicher Einstellungen für Ihren ActiveMQ-Broker finden Sie unter Erste Schritte: Einen ActiveMQ-Broker erstellen und eine Verbindung zu ihm herstellen.

Löschen eines Amazon MQ Cross-Region-Datenreplikations-Brokers

Um einen primären Broker oder einen Replikatbroker für die regionsübergreifende Datenreplikation (CRDR) zu löschen, müssen Sie die Broker zuerst entkoppeln und dann neu starten. Die folgenden Anweisungen zeigen, wie Sie die Broker mithilfe der Management Console entkoppeln und neu starten können. AWS

1. Wählen Sie auf der Broker-Seite den CRDR Broker aus, den Sie entkoppeln möchten, und klicken Sie dann auf Bearbeiten.
2. Wählen Sie auf der Seite Bearbeiten des Brokers im Abschnitt Datenreplikation die Option Broker entkoppeln aus.

3. Geben Sie im Popup-Fenster „Entkoppeln“ ein, um Ihre Auswahl zu bestätigen. Wählen Sie dann Broker entkoppeln aus.
4. Starten Sie anschließend den entkoppelten Primär-Broker neu. Dadurch wird auch der Replikat-Broker neu gestartet. Anleitungen zum Neustart Ihres Brokers finden Sie unter Rebooting a Broker. Nach dem Neustart des Primär-Brokers sind beide Broker entkoppelt und können einzeln gelöscht werden. Informationen zum Löschen Ihres Brokers finden Sie unter Deleting a broker.

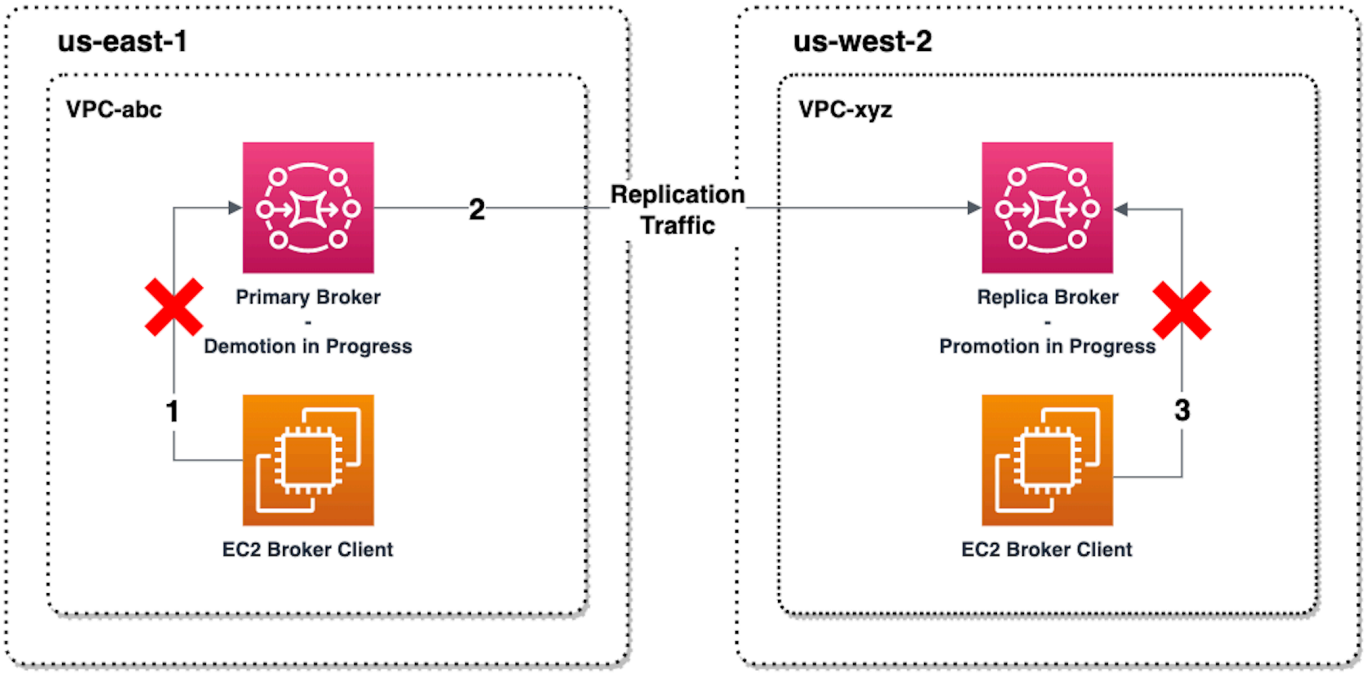
Initiieren eines Switchovers oder Failovers, um einen Amazon MQ MQ-Replikatbroker zur Rolle des primären Brokers hochzustufen

Sie können ein Switchover oder Failover initiieren, wenn Sie den Replikat-Broker in die Rolle des Primär-Brokers hochstufen möchten. Wenn Sie den Replikat-Broker hochstufen, wird der Primär-Broker in die Rolle des Replikat-Brokers heruntergestuft.

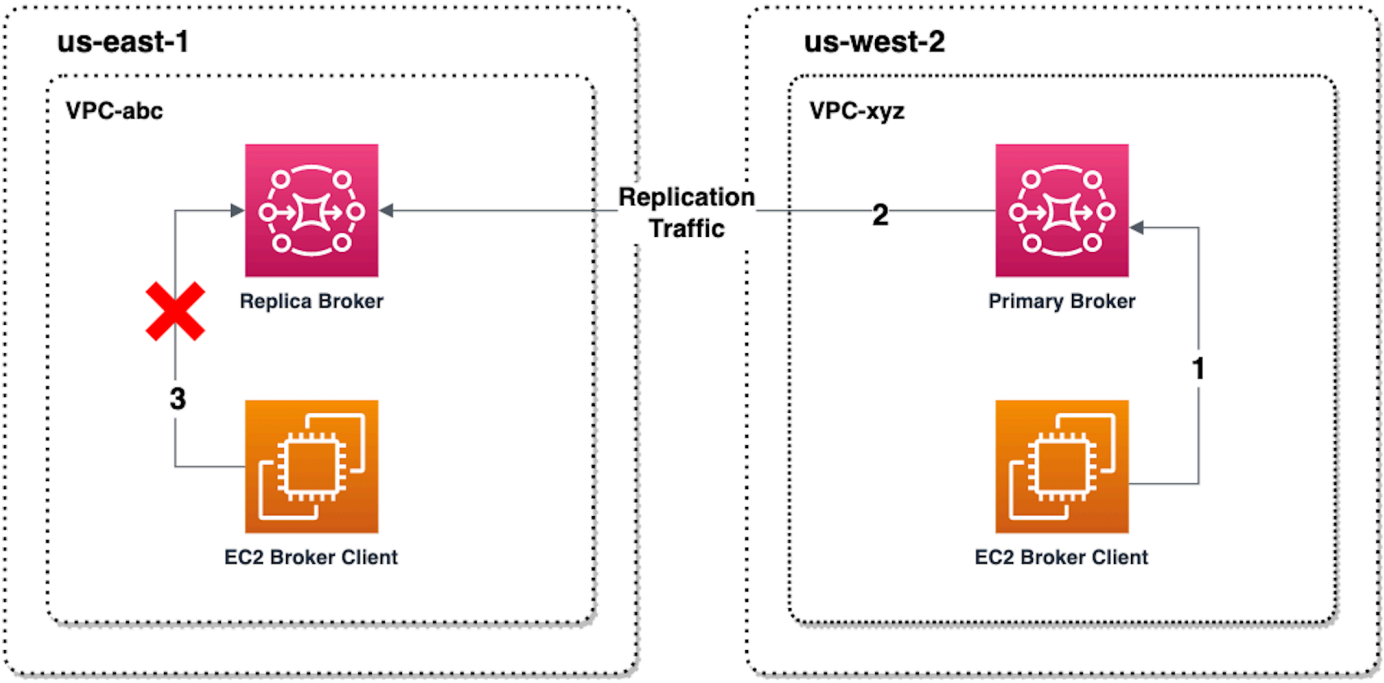
Bei einem Switchover hat die Konsistenz Vorrang vor der Verfügbarkeit. Die Broker haben garantiert den gleichen Status, wenn der Failover-Vorgang abgeschlossen ist. Bei einem Switchover kann es einen Zeitraum geben, in dem keiner der beiden Broker für Clientverbindungen verfügbar ist, während die Konsistenz zwischen den Brokern hergestellt wird. Beide Broker haben den gleichen Status, wenn das Replikat hochgestuft wird. Der Erfolg des Switchover hängt vom Zustand beider Regionen und des regionsübergreifenden Netzes ab.

Bei einem Failover hat die Verfügbarkeit Vorrang vor der Konsistenz. Es kann nicht garantiert werden, dass Makler nach Abschluss dieses Vorgangs identische Status haben. Bei einem Failover ist der Replikat-Broker garantiert sofort verfügbar, um den Client-Datenverkehr zu bearbeiten, ohne auf die Synchronisierung der Replikationsdaten oder auf das Signal zum Herunterfahren des Primär-Brokers zu warten. Der Erfolg des Failovers hängt weder vom Zustand der ursprünglichen primären Region noch vom Netzwerk zwischen den Regionen ab.

Das folgende Diagramm veranschaulicht einen Switchover, bei dem keiner der beiden Broker Clientverbindungen annimmt, während die Replikationswarteschlange geleert wird und die Broker-Status synchronisiert werden. Bei diesem Vorgang VPC ist der Client im primären Broker nicht in der Lage, während des Vorgangs weitere Statusänderungen vorzunehmen, und der primäre Broker wird zu einem Replikat herabgestuft. Wenn die Replikationswarteschlange leer ist und die beiden Broker den gleichen Status erreichen, kann der Client im Replikatbroker keine Verbindung zum Replikatbroker herstellen, bis der Failover-Vorgang abgeschlossen VPC ist und der Replikatbroker zum primären Broker heraufgestuft wird.



Das folgende Diagramm veranschaulicht den Broker-Status, nachdem der Switchover-Vorgang abgeschlossen ist. Der ursprüngliche Replikat-Broker wurde zum Primär-Broker hochgestuft und nimmt nun Clientverbindungen an. Der Client kann Daten vom Broker erstellen und verwenden.



Hochstufen des Replikat-Brokers über die Konsole

Führen Sie die folgenden Schritte in der Amazon-MQ-Konsole aus, um den Replikat-Broker mittels Switchover oder Failover hochzustufen.

> **Note**
>
> Sie können weder ein Switchover noch ein Failover auf einem Primär-Broker initiieren.

1. Wechseln Sie zu der Region für Ihren Replica-Broker. Wählen Sie in der Broker-Tabelle den vorhandenen Replikat-Broker aus, den Sie zu einem Primär-Broker hochstufen möchten.
2. Führen Sie auf der Seite mit Broker-Details Folgendes aus:
    1. Wählen Sie Replikat hochstufen aus.
    2. Wählen Sie im Popup-Fenster Switchover oder Failover aus.
    3. Geben Sie „Bestätigen“ in das Textfeld ein, um Ihre Auswahl zu bestätigen.
    4. Wählen Sie Bestätigen aus.

Nach dem Initiieren des Failovers ändert sich der Broker-Status in Failover läuft. Der blaue Fortschrittsbalken oben auf der Seite „Broker“ wechselt zu grün, wenn der Failover-Vorgang abgeschlossen ist.

> **Note**
>
> Die Konfiguration wird nur zum Zeitpunkt der Replikat-Broker-Erstellung repliziert. Jedes nachfolgende Update wird nicht repliziert.

Regionsübergreifende Datenreplikationsmetriken in Amazon CloudWatch

Die Funktion der regionsübergreifenden Datenreplikation von Amazon MQ für ActiveMQ bietet Metriken zur Aufrechterhaltung der Zuverlässigkeit, Verfügbarkeit und Leistung Ihrer Primär- und Replikat-Broker. Während des Replikationsprozesses empfängt ein Replikat-Broker in einer sekundären Region asynchron replizierte Daten von dem Primär-Broker in der primären Region. Wenn der Primär-Broker in der primären Region ausfällt, können Sie den Replikat-Broker in der sekundären Region zum Primär-Broker hochstufen, indem Sie ein Switchover oder Failover einleiten.

Anweisungen zum Anzeigen von Metriken in Amazon CloudWatch finden Sie unter Zugreifen auf CloudWatch Metriken für Amazon MQ.

CRDR-Zeitstempel

Die folgenden Zeitstempel beschreiben, wie die Metriken in Amazon CloudWatch Metriken berechnet werden. Beim Datenreplikationsprozess gibt es fünf Zeitstempel:

- Zeitpunkt der aktuellen Beobachtung (TCO): Der aktuelle Zeitpunkt.
- Zeitpunkt der Erstellung (TC): Der Zeitpunkt, zu dem ein Ereignis vom Primär-Broker in der Replikationswarteschlange erstellt wurde. Verfügbar für Primär- und Replikat-Broker.
- Zeitpunkt der Zustellung (TD): Der Zeitpunkt, zu dem ein Ereignis erfolgreich an den Replikat-Broker übermittelt wurde. Nur auf Replikat-Brokern verfügbar.
- Zeitpunkt der Bearbeitung (TP): Der Zeitpunkt, zu dem ein Ereignis vom Replikat-Broker erfolgreich verarbeitet wurde. Nur auf Replikat-Brokern verfügbar.
- Zeitpunkt der Bestätigung (TA): Der Zeitpunkt, zu dem ein Ereignis erfolgreich vom Primär-Broker bestätigt wurde. Nur auf Primär-Brokern verfügbar.

Einschätzen der Switchover/Failover-Leistung mit CRDR-CloudWatch-Metriken

Amazon MQ aktiviert standardmäßig Metriken für Ihren Broker. Sie können Ihre Broker-Metriken anzeigen, indem Sie auf die Amazon-CloudWatch-Konsole zugreifen oder die CloudWatch-API verwenden. Die folgenden Metriken sind hilfreich, um die Replikations- und Switchover/Failover-Leistung Ihrer CRDR-Broker zu verstehen:

Amazon–MQ-CloudWatch-Metrik	Grund für die Verwendung von CRDR	
`TotalReplicationLag`	Die geschätzte Zeit zwischen TA und TC des letzten unbestätigten Ereignisses auf dem Primär-Broker.	
`ReplicationLag`	Die geschätzte Zeit zwischen TP und TC des letzten unbestätigten Ereignisses auf dem Replikat-Broker.	

Amazon–MQ-CloudWatch-Metrik	Grund für die Verwendung von CRDR	
`PrimaryWaitTime`	Die geschätzte Zeit zwischen TCO und TC des letzten bearbeiteten Ereignisses auf dem Primär-Broker.	
`ReplicaWaitTime`	Die geschätzte Zeit zwischen TCO und TP des zuletzt bearbeiteten Ereignisses auf dem Replikat-Broker.	
`QueueSize`	Die Gesamtzahl der unbestätigten Ereignisse in der Replikationswarteschlange auf dem Primär-Broker.	

`TotalReplicationLag` und `ReplicationLag` beschreiben die verzögerte Replikation zwischen dem Primär-und dem Replikat-Broker. Die beiden Metriken können auch verwendet werden, um die Zeit bis zum Abschluss des laufenden Switchover- oder Failover-Vorgangs abzuschätzen.

`PrimaryWaitTime` und `ReplicaWaitTime` können verwendet werden, um alle laufenden Probleme mit dem Replikationsprozess zu identifizieren. Wenn der Wert der Metrik ständig steigt, kann dies darauf hindeuten, dass der Replikationsprozess beeinträchtigt oder unterbrochen wurde. Aufgrund von Problemen wie der Netzwerkpartitionierung, Brokerstarts und einer langen Wiederherstellung kann es zu einer langsamen Replikation kommen.

ActiveMQ Tutorials

Die folgenden Tutorials zeigen, wie Sie Ihre ActiveMQ-Broker erstellen und eine Verbindung mit ihnen herstellen können. Wenn Sie den ActiveMQ Java Beispiel-Code verwenden möchten, müssen Sie das Java Standard Edition Development Kit installieren und einige Konfigurationsänderungen am Code vornehmen.

Themen

- Erstellen und Konfigurieren eines Amazon MQ-Netzwerks von Brokern

- Verbinden einer Java-Anwendung mit Ihrem Amazon MQ-Broker
- Integration von ActiveMQ-Brokern in LDAP
- Einen ActiveMQ-Broker-Benutzer erstellen
- Einen ActiveMQ-Broker-Benutzer bearbeiten
- Löschen Sie einen ActiveMQ-Broker-Benutzer
- Arbeitsbeispiele für die Verwendung von Java Message Service (JMS) mit ActiveMQ

Erstellen und Konfigurieren eines Amazon MQ-Netzwerks von Brokern

Ein -Netzwerk von Brokernbesteht aus mehreren gleichzeitig aktivenSingle-Instance-Brokernoderaktiven/Standby-Brokern. Sie können Brokernetzwerke in einer Vielzahl von Topologien (z. B. Concentrator hub-and-spokes, Tree oder Mesh) konfigurieren, je nach den Anforderungen Ihrer Anwendung, z. B. Hochverfügbarkeit und Skalierbarkeit. Zum Beispiel kann ein Hub-und-Spoke-Netzwerk von Brokern kann die Ausfallsicherheit erhöhen und Nachrichten erhalten, wenn ein Broker nicht erreichbar ist. Ein Netzwerk von Brokern mit einemKonzentratorTopologie kann Nachrichten von einer größeren Anzahl von Brokern sammeln, die eingehende Nachrichten akzeptieren, und sie auf zentralere Broker konzentrieren, um die Belastung vieler eingehender Nachrichten besser zu bewältigen. In diesem Tutorial erfahren Sie, wie Sie ein Zwei-Broker-Netzwerk von Brokern mit einer Source and Sink-Topologie erstellen.

Eine konzeptionelle Übersicht und detaillierte Konfigurationsinformationen finden Sie im Folgenden:

- Amazon MQ Brokernetzwerk
- Korrekte Konfiguration Ihres Netzwerk von Brokern
- `networkConnector`
- `networkConnectionStartAsynchron`
- Netzwerke von Brokern in der ActiveMQ-Dokumentation

Sie können die Amazon MQ Konsole verwenden, um ein Amazon MQ-Netzwerk von Brokern zu erstellen. Da Sie die Erstellung der beiden Broker parallel starten können, dauert dieser Prozess ca. 15 Minuten.

Themen

- Voraussetzungen
- Schritt 1: Zulassen von Datenverkehr zwischen Brokern

- Schritt 2: Konfigurieren von Netzwerk-Connectors für Ihren Broker
- Nächste Schritte

Voraussetzungen

Um ein Netzwerk von Brokern zu erstellen, müssen Sie über Folgendes verfügen:

- Zwei oder mehr gleichzeitig aktive Broker (in diesem Tutorial `MyBroker1` und `MyBroker2` genannt). Weitere Informationen zum Erstellen von Brokern finden Sie unter Erste Schritte: Einen ActiveMQ-Broker erstellen und eine Verbindung zu ihm herstellen.
- Die beiden Broker müssen sich im selben Netzwerk VPC oder im Peering-Modus befinden. VPCs Weitere Informationen zu VPCs finden Sie unter Was ist AmazonVPC? im VPCAmazon-Benutzerhandbuch und Was ist VPC Peering? im Amazon VPC Peering Guide.

  > **Important**
  >
  > Wenn Sie keine StandardwerteVPC, Subnetze oder Sicherheitsgruppen haben, müssen Sie diese zuerst erstellen. Weitere Informationen finden Sie im VPCAmazon-Benutzerhandbuch:
  >
  > - Eine Standardeinstellung erstellen VPC
  > - Erstellen eines Standard-Subnetzes
  > - Erstellen einer Sicherheitsgruppe

- Zwei Benutzer mit identischen Anmeldeinformationen für beide Broker. Weitere Informationen zum Erstellen von Benutzern finden Sie unter Einen ActiveMQ-Broker-Benutzer erstellen.

  > **Note**
  >
  > Stellen Sie bei der Integration der LDAP Authentifizierung in ein Brokernetzwerk sicher, dass der Benutzer sowohl als ActiveMQ-Broker als auch als LDAP Benutzer existiert.

Das folgende Beispiel verwendet zwei Single-Instance-Broker. Sie können jedoch Netzwerke von Brokern mit Hilfe von aktiv/standby-Brokern oder einer Kombination von Broker-Bereitstellungsarten erstellen.

Schritt 1: Zulassen von Datenverkehr zwischen Brokern

Nachdem Sie Ihre Broker erstellt haben, müssen Sie den Datenverkehr zwischen ihnen zulassen.

1. Wählen Sie auf der Amazon MQ MQ-Konsole auf der MyBroker2-Seite im Abschnitt Details unter Sicherheit und Netzwerk den Namen Ihrer Sicherheitsgruppe oder

   Die Seite „Sicherheitsgruppen“ des EC2 Dashboards wird angezeigt.
2. Wählen Sie in der Liste der Sicherheitsgruppen Ihre Sicherheitsgruppe.
3. Klicken Sie unten auf der Seite auf Inbound (Eingehend) und anschließend auf Edit (Bearbeiten).
4. Fügen Sie im Dialogfeld Regeln für eingehenden Datenverkehr bearbeiten eine Regel für den OpenWire Endpunkt hinzu.

   a. Klicken Sie auf Add Rule (Regel hinzufügen).

   b. Wählen Sie für Typ die Option Benutzerdefiniert TCP aus.

   c. Geben Sie für Portbereich den OpenWire Port (`61617`) ein.

   d. Führen Sie eine der folgenden Aktionen aus:

   - Wenn Sie den Zugriff auf eine bestimmte IP-Adresse einschränken möchten, lassen Sie bei Source (Quelle), Custom (Benutzerdefiniert) ausgewählt, und geben Sie dann die IP-Adresse von `MyBroker1` gefolgt von `/32` ein. (Dadurch wird die IP-Adresse in einen gültigen CIDR Datensatz umgewandelt). Weitere Informationen finden Sie unter Elastic Network Interfaces (Elastic Network-Schnittstellen).

     > **Tip**
     >
     > Wählen Sie zum Abrufen der IP-Adresse von `MyBroker1` in der Amazon MQ-Konsole den Namen des Brokers aus und navigieren Sie zum Abschnitt Details.

   - Wenn alle Broker privat sind und demselben Unternehmen angehörenVPC, lassen Sie für Quelle die Option Benutzerdefiniert ausgewählt und geben Sie dann die ID der Sicherheitsgruppe ein, die Sie bearbeiten.

> **Note**
>
> Für öffentliche Broker müssen Sie den Zugriff unter Verwendung von IP-Adressen einschränken.

e. Wählen Sie Save (Speichern) aus.

Ihr Broker kann nun eingehende Verbindungen akzeptieren.

Schritt 2: Konfigurieren von Netzwerk-Connectors für Ihren Broker

Nachdem Sie den Datenverkehr zwischen Ihren Brokern zugelassen haben, müssen Sie Netzwerk-Connectors für einen von ihnen konfigurieren.

1. Bearbeiten Sie die Konfigurationsrevision für den Broker `MyBroker1`.

   a. Wählen Sie auf der Seite MyBroker1 die Option Bearbeiten aus.

   b. Wählen Sie auf der Seite Edit MyBroker 1 im Abschnitt Konfiguration die Option View aus.

   Der Typ der Broker-Engine und die Version, die die Konfiguration verwendet (z. B. Apache ActiveMQ 5.15.0) werden angezeigt.

   c. Auf der Registerkarte „Konfigurationsdetails“ werden die Versionsnummer, die Beschreibung und das XML Format der Brokerkonfiguration angezeigt.

   d. Wählen Sie Edit configuration (Konfiguration bearbeiten) aus.

   e. Entkommentieren Sie am Ende der Konfigurationsdatei den Abschnitt `<networkConnectors>` und fügen Sie die folgenden Informationen hinzu:

   - Den `name` für den Netzwerk-Connector.
   - Die ActiveMQ-Webkonsolen-`username` der beiden Brokern gemeinsam ist.
   - Aktivieren Sie `duplex`-Verbindungen.
   - Führen Sie eine der folgenden Aktionen aus:
     - Wenn Sie den Broker mit einem Single-Instance-Broker verbinden, verwenden Sie das `static:` Präfix und den OpenWire Endpunkt `uri` für`MyBroker2`. Beispielsweise:

```
<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>
```

- Wenn Sie den Broker mit einem aktiven/Standby-Broker verbinden, verwenden Sie den `static+failover` Transport und den OpenWire Endpunkt `uri` für beide Broker mit den folgenden Abfrageparametern. `?randomize=false&maxReconnectAttempts=0` Beispielsweise:

```
<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&amp;maxReconnectAttempts=0)"/>
</networkConnectors>
```

> **Note**
>
> Geben Sie die Anmeldeinformationen für den ActiveMQ-Benutzer nicht an.

f. Wählen Sie Save (Speichern) aus.

g. Geben Sie im Dialogfeld Save revision (Revision speichern) `Add network of brokers connector for MyBroker2` ein.

h. Wählen Sie Save (Speichern) aus, um die neue Revision der Konfiguration zu speichern.

2. Bearbeiten Sie `MyBroker1`, um die neueste Revision der Konfiguration so einzustellen, dass sie sofort wirksam wird.

a. Wählen Sie auf der Seite MyBroker1 die Option Bearbeiten aus.

b. Wählen Sie auf der Seite Bearbeiten MyBroker 1 im Abschnitt Konfiguration die Option Änderungen planen aus.

c. Wählen Sie im Abschnitt Schedule broker modifications (Broker-Änderungen planen) aus, dass Änderungen Immediately (Sofort) wirksam werden sollen.

d. Wählen Sie Apply (Anwenden) aus.

`MyBroker1` wird neu gestartet und Ihre Konfigurationsrevision wird angewendet.

Das Netzwerk von Brokern wird erstellt.

Nächste Schritte

Nachdem Sie Ihr Netzwerk von Brokern konfiguriert haben, können Sie es testen, indem Sie Nachrichten produzieren und konsumieren.

> **Important**
>
> Stellen Sie sicher, dass Sie eingehende Verbindungen von Ihrem lokalen Computer für den Broker `MyBroker1` auf Port 8162 (für die ActiveMQ Web Console) und Port 61617 (für den Endpunkt) aktivieren. OpenWire
> Möglicherweise müssen Sie auch die Einstellungen Ihrer Sicherheitsgruppe(n) anpassen, damit der Produzent und der Verbraucher eine Verbindung zum Netzwerk der Broker herstellen können.

1. Navigieren Sie in der Amazon MQ-Konsole zum Abschnitt Connections (Verbindungen) und notieren Sie sich den ActiveMQ Web Console-Endpunkt für den Broker `MyBroker1`.
2. Navigieren Sie zur ActiveMQ Web Console für den Broker `MyBroker1`.
3. Um zu überprüfen, ob die Netzwerkbrücke verbunden ist, wählen Sie Network (Netzwerk) aus.

   Im Abschnitt Network Bridges (Netzwerkbrücken) werden der Name und die Adresse von `MyBroker2` in den Spalten Remote Broker (Remote-Broker) und Remote Address (Remote-Adresse) aufgeführt.
4. Erstellen Sie von einem beliebigen Computer mit Zugriff auf den Broker `MyBroker2` einen Verbraucher. Beispielsweise:

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

Der Verbraucher stellt eine Verbindung zum OpenWire Endpunkt von her `MyBroker2` und beginnt, Nachrichten aus der Warteschlange zu konsumieren. `MyQueue`

5. Erstellen Sie von einem beliebigen Computer mit Zugriff auf den Broker `MyBroker1` einen Produzenten und senden Sie einige Nachrichten. Beispielsweise:

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

Der Producer stellt eine Verbindung zum OpenWire Endpunkt von her `MyBroker1` und beginnt, persistente Nachrichten für die Warteschlange zu erzeugen`MyQueue`.

Verbinden einer Java-Anwendung mit Ihrem Amazon MQ-Broker

Nachdem Sie einen Amazon MQ ActiveMQ Broker erstellt haben, können Sie Ihre Anwendung mit ihm verbinden. Die folgenden Beispiele zeigen, wie Sie den Java Message Service (JMS) verwenden können, um eine Verbindung zum Broker herzustellen, eine Warteschlange zu erstellen und eine Nachricht zu senden. Ein vollständiges, funktionierendes Java-Beispiel finden Sie unterWorking Java Example.

Sie können unter Verwendung verschiedener ActiveMQ-Clients eine Verbindung zu ActiveMQ-Brokern einrichten. Wir empfehlen die Verwendung des ActiveMQ-Clients.

Themen

- Voraussetzungen
- So erstellen Sie einen Nachrichtenproduzenten und senden eine Nachricht:
- So erstellen Sie einen Nachrichtenkonsumenten und empfangen die Nachricht:

Voraussetzungen

VPCAttribute aktivieren

Um sicherzustellen, dass Ihr Broker in Ihrem erreichbar istVPC, müssen Sie die `enableDnsSupport` VPC Attribute `enableDnsHostnames` und aktivieren. Weitere Informationen finden Sie unter DNSSupport VPC in Ihrem VPC Amazon-Benutzerhandbuch.

Eingehende Verbindungen aktivieren

Aktivieren Sie als Nächstes eingehende Verbindungen für Ihre Anwendung.

1. Melden Sie sich bei der Amazon-MQ-Konsole an.
2. Wählen Sie aus der Brokerliste den Namen Ihres Brokers aus (z. B. MyBroker).
3. Auf der ***MyBroker***Notieren Sie sich im Abschnitt Verbindungen die Adressen und Ports der Webkonsole URL und der Wire-Level-Protokolle des Brokers.
4. Wählen Sie im Abschnitt Details unter Sicherheit und Netzwerk den Namen Ihrer Sicherheitsgruppe oder

   .

   Die Seite „Sicherheitsgruppen“ des EC2 Dashboards wird angezeigt.
5. Wählen Sie in der Liste der Sicherheitsgruppen Ihre Sicherheitsgruppe.
6. Klicken Sie unten auf der Seite auf Inbound (Eingehend) und anschließend auf Edit (Bearbeiten).
7. Fügen Sie im Dialogfeld „Regeln für eingehenden Datenverkehr bearbeiten“ eine Regel für jeden URL oder Endpunkt hinzu, auf den Sie öffentlich zugreifen möchten (das folgende Beispiel zeigt, wie Sie dies für eine Broker-Webkonsole tun können).

   a. Klicken Sie auf Add Rule (Regel hinzufügen).

   b. Wählen Sie für Typ die Option Benutzerdefiniert TCP aus.

   c. Für Port-Bereich, geben Sie den Port der Webkonsole ein (`8162`).

   d. Für Source (Quelle), lassen Sie Custom (Benutzerdefiniert) ausgewählt, und geben Sie dann die IP-Adresse des Systems ein, auf das auf die Webkonsole zugegriffen werden soll (z. B. `192.0.2.1`) enthalten.

   e. Wählen Sie Save.

   Ihr Broker kann nun eingehende Verbindungen akzeptieren.

Java-Abhängigkeiten hinzufügen

Fügen Sie dem Pfad für Ihre Java-Build-Klasse die Pakete `activemq-client.jar` und `activemq-pool.jar`hinzu. Das folgende Beispiel zeigt diese Abhängigkeiten in der `pom.xml`-Datei eines Maven-Projekts.

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

Weitere Informationen über `activemq-client.jar` finden Sie unter Ursprüngliche Konfiguration in der Apache ActiveMQ-Dokumentation.

> **Important**
>
> Im folgenden Beispielcode laufen Hersteller und Verbraucher in einem einzigen Thread. Stellen Sie für Produktionssysteme (oder zum Testen des Failovers von Broker-Instances) sicher, dass Ihre Produzenten und Verbraucher auf separaten Hosts oder Threads ausgeführt werden.

So erstellen Sie einen Nachrichtenproduzenten und senden eine Nachricht:

Verwenden Sie die folgende Anweisung, um einen Nachrichtengenerator zu erstellen und eine Nachricht zu empfangen.

1. Erstellen Sie mithilfe des Endpunkts Ihres Brokers eine JMS gepoolte Verbindungs-Factory für den Nachrichtenproduzenten und rufen Sie dann die `createConnection` Methode für die Factory auf.

> **Note**
>
> Für einen Aktiv-/Standby-Broker bietet Amazon MQ zwei ActiveMQ-Web-KonsolenURLs, von denen jedoch jeweils nur eine aktiv URL ist. Ebenso stellt Amazon MQ zwei Endpunkte für jedes Wire-Level-Protokoll bereit, jedoch ist jeweils nur ein Endpunkt in jedem Paar aktiv. Die `-1`- und `-2`-Suffixe bezeichnen ein redundantes Paar. Weitere Informationen finden Sie unter Bereitstellungsoptionen für Amazon MQ für ActiveMQ-Broker).
>
> Für Drahtebene Protokollendpunkte können Sie zulassen, dass Ihre Anwendung eine Verbindung zu einem beliebigen Endpunkt herstellen kann, indem Sie dieFailover-Transport verwenden.

```
// 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**
>
> Nachrichtenproduzenten sollten immer die`PooledConnectionFactory`-Klasse. Weitere Informationen finden Sie unter Verwenden Sie immer Verbindungspools.

2. Erstellen Sie eine Sitzung, eine Warteschlange namens`MyQueue`und einen Nachrichtenproduzenten.

```
// 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. Erstellen der Nachrichtenzeichenfolge`"Hello from Amazon MQ!"`Dann senden Sie die Nachricht.

```
// 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. Bereinigen Sie den Produzenten.

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

So erstellen Sie einen Nachrichtenkonsumenten und empfangen die Nachricht:

Verwenden Sie die folgende Anweisung, um einen Nachrichtenproduzenten zu erstellen und eine Nachricht zu empfangen.

1. Erstellen Sie mithilfe des Endpunkts Ihres Brokers eine JMS Verbindungs-Factory für den Nachrichtengenerator und rufen Sie dann die `createConnection` Methode für die Factory auf.

```
// 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**
>
> Die Nachrichtenkonsumenten sollten nie die `PooledConnectionFactory`-Klasse verwenden. Weitere Informationen finden Sie unter Verwenden Sie immer Verbindungspools.

2. Erstellen Sie eine Sitzung, eine Warteschlange namens`MyQueue`und einem Nachrichtenverbraucher.

```
// 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. Beginnen Sie, auf Nachrichten zu warten und die Nachricht zu erhalten, wenn sie eintrifft.

```
// 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**
>
> Im Gegensatz zu AWS Messaging-Diensten (wie AmazonSQS) ist der Verbraucher ständig mit dem Broker verbunden.

4. Schließen Sie den Verbraucher, die Sitzung und die Verbindung.

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

Integration von ActiveMQ-Brokern in LDAP

> **Important**
>
> Die LDAP-Integration wird für RabbitMQ-Broker nicht unterstützt.

Sie können über die folgenden Protokolle mit aktiviertem TLS auf Ihre ActiveMQ-Broker zugreifen:

- AMQP
- MQTT
- MQTT über WebSocket
- OpenWire
- STOMP
- STOMP über WebSocket

Amazon MQ bietet die Wahl zwischen nativer ActiveMQ-Authentifizierung und LDAP-Authentifizierung und -Autorisierung, um Benutzerberechtigungen zu verwalten. Weitere Informationen über Einschränkungen im Zusammenhang mit ActiveMQ-Benutzernamen und -Passwörtern finden Sie unter Benutzer.

Um ActiveMQ-Benutzer und -Gruppen für die Arbeit mit Warteschlangen und Themen zu autorisieren, müssen Sie die Konfiguration Ihres Brokers bearbeiten. Amazon MQ verwendet zum Einschränken des Lese- und Schreibzugriffs auf Ziele das Simple Authentication Plugin von ActiveMQ. Weitere

Informationen und Beispiele finden Sie unter Immer eine Autorisierungszuordnung konfigurieren und `authorizationEntry`.

> **Note**
>
> Derzeit unterstützt Amazon MQ keine Clientzertifikat-Authentifizierung.

Themen

Integrieren von LDAP mit ActiveMQ

Sie können Amazon MQ Benutzer über die Anmeldeinformationen authentifizieren, die in Ihrem LDAP-Server (Lightweight Directory Access Protocol) gespeichert sind. Außerdem können Sie Amazon-MQ-Benutzer hinzufügen, löschen und ändern und Themen und Warteschlangen Berechtigungen zuweisen. Verwaltungsvorgänge wie das Erstellen, Aktualisieren und Löschen von Brokern erfordern weiterhin IAM-Anmeldeinformationen und sind nicht in LDAP integriert.

Kunden, die ihre Amazon-MQ-Broker-Authentifizierung und -Autorisierung mithilfe eines LDAP-Servers vereinfachen und zentralisieren möchten, können diese Funktion nutzen. Das Speichern aller Benutzeranmeldeinformationen auf dem LDAP-Server spart Zeit und Aufwand, da ein zentraler Speicherort für die Speicherung und Verwaltung dieser Anmeldeinformationen bereitgestellt wird.

Amazon MQ bietet LDAP-Unterstützung mit dem Apache-ActiveMQ-JAAS-Plugin. Alle vom Plugin unterstützten LDAP-Server wie Microsoft Active Directory oder OpenLDAP werden ebenfalls von Amazon MQ unterstützt. Weitere Informationen zum Plugin finden Sie unter dem Abschnitt Sicherheit in der Active-MQ-Dokumentation.

Zusätzlich zu Benutzern können Sie den Zugriff auf Themen und Warteschlangen für eine bestimmte Gruppe oder einen Benutzer über Ihren LDAP-Server festlegen. Dazu erstellen Sie Einträge, die Themen und Warteschlangen auf Ihrem LDAP-Server darstellen und dann Berechtigungen einem bestimmten LDAP-Benutzer oder einer Gruppe zuweisen. Anschließend können Sie den Broker so konfigurieren, dass er Autorisierungsdaten vom LDAP-Server abruft.

Voraussetzungen

Bevor Sie LDAP-Support zu einem neuen oder vorhandenen Amazon-MQ-Broker hinzufügen, müssen Sie ein Service-Konto einrichten. Dieses Servicekonto ist erforderlich, um eine Verbindung zu einem LDAP-Server herzustellen und muss über die richtigen Berechtigungen verfügen, um diese Verbindung herzustellen. Dieses Dienstkonto richtet die LDAP-Authentifizierung für Ihren Broker ein. Alle aufeinanderfolgenden Clientverbindungen werden über dieselbe Verbindung authentifiziert.

Ein Servicekonto ist ein Konto auf Ihrem LDAP-Server, das eine Verbindung initiieren kann. Es handelt sich um eine standardmäßige LDAP-Anforderung, und Sie müssen die Anmeldeinformationen des Servicekontos nur einmal angeben. Nachdem die Verbindung eingerichtet wurde, werden alle zukünftigen Clientverbindungen über Ihren LDAP-Server authentifiziert. Ihre Anmeldeinformationen für das Dienstkonto werden sicher in verschlüsselter Form gespeichert, auf die nur Amazon MQ zugegriffen werden kann.

Für die Integration mit ActiveMQ ist eine bestimmte Directory Information Tree (DIT) auf dem LDAP-Server erforderlich. Eine beispielshafte `ldif`-Datei, die diese Struktur deutlich zeigt, finden Sie unterImportieren Sie die folgende LDIF-Datei in den LDAP-Server im Abschnitt Sicherheit in der ActiveMQ-Dokumentation.

Erste Schritte mit LDAP

Um zu beginnen, navigieren Sie zur Amazon MQ Konsole und wählen Sie LDAP-Authentifizierung und -Autorisierung, wenn Sie eine neue Amazon MQ erstellen oder eine vorhandene Broker-Instance bearbeiten.

Geben Sie die folgenden Informationen zum Servicekonto ein:

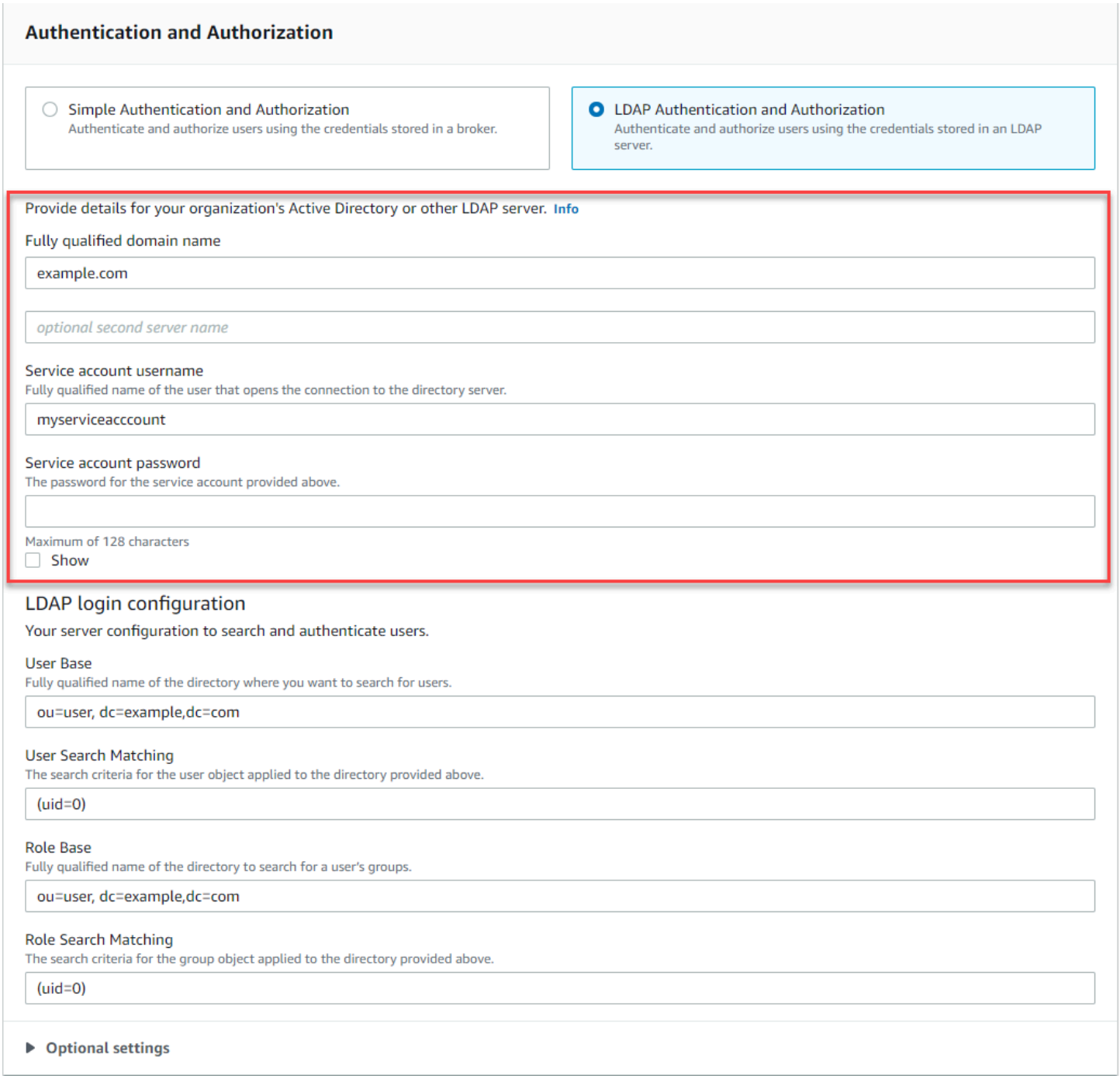
- Vollqualifizierter Domänenname Der Speicherort des LDAP-Servers, an den Authentifizierungs- und Autorisierungsanforderungen ausgegeben werden sollen.

> **Note**
>
> Der vollqualifizierte Domänenname des von Ihnen angegebenen LDAP-Servers darf nicht das Protokoll oder die Portnummer enthalten. Amazon MQ wird dem vollqualifizierten Domänennamen das Protokoll `ldaps` vorangestellt, und fügt die Portnummer `636` hinzu. Wenn Sie beispielsweise die folgende vollqualifizierte Domäne angeben: `example.com`, greift Amazon MQ über die folgende URL auf Ihren LDAP-Server zu: `ldaps://example.com:636`.

Damit der Brokerhost erfolgreich mit dem LDAP-Server kommunizieren kann, muss der vollqualifizierte Domänenname öffentlich aufgelöst werden. Um den LDAP-Server privat und sicher zu halten, beschränken Sie den eingehenden Datenverkehr in den eingehenden Regeln des Servers, so dass nur Datenverkehr zugelassen wird, der aus der VPC des Brokers stammt.

- Benutzername für Service-Konto Der definierte Name des Benutzers, der verwendet wird, um die anfängliche Bindung an den LDAP-Server durchzuführen.
- Passwort des Service-Kontos Das Passwort des Benutzers, der die anfängliche Bindung ausführt.

In der folgenden Abbildung wird hervorgehoben, wo diese Details angegeben werden sollen.

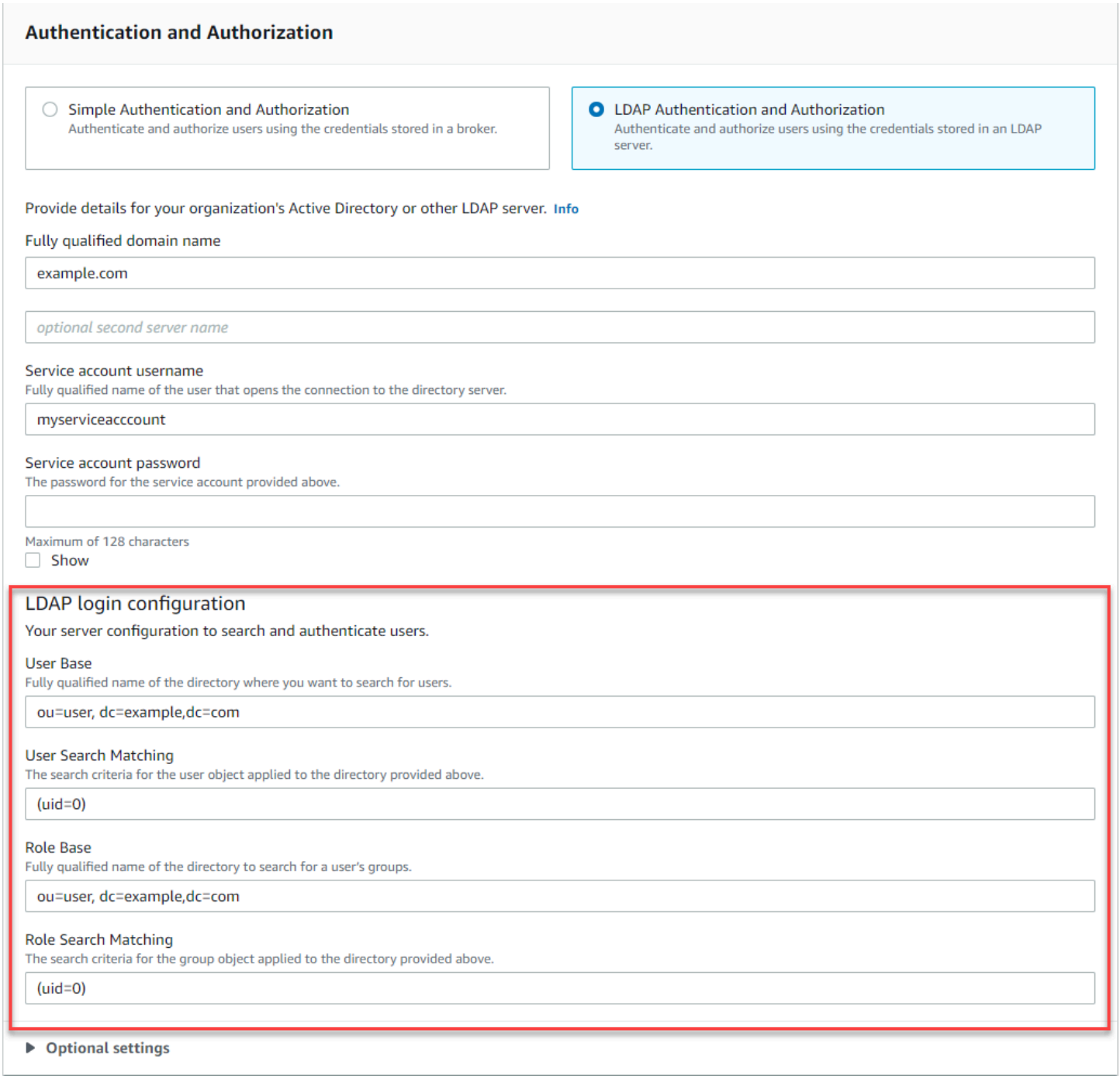


In der Konfiguration der LDAP-Anmeldung geben Sie die folgenden erforderlichen Informationen ein:

- Benutzerbasis Der definierte Name des Knotens im Directory Information Tree (DIT, Verzeichnisinformationsbaum), der nach Benutzern durchsucht werden soll.
- Benutzer-Suchabgleich Der LDAP-Suchfilter, der für die Suche nach Benutzern innerhalb der `userBase` verwendet wird. Der Benutzername des Kunden wird im Suchfilter mit dem Platzhalter `{0}` ersetzt. Weitere Informationen finden Sie unter Authentifizierung und Autorisierung.

- Rollenbasis Der definierte Name des Knotens im DIT, der nach Rollen durchsucht werden soll. Rollen können als explizite LDAP-Gruppeneinträge in Ihrem Verzeichnis konfiguriert werden. Ein typischer Rolleneintrag kann aus einem Attribut für den Namen der Rolle bestehen, z. B. common name (CN, allgemeiner Name) und ein anderes Attribut, wie `member`, mit Werten, die die definierten Namen oder Benutzernamen der Benutzer der Rollengruppe darstellen. Zum Beispiel, angesichts der Organisationseinheit, `group`, können Sie den folgenden definierten Namen angeben: `ou=group,dc=example,dc=com`.
- Rollen-Suchabgleich Der LDAP-Suchfilter, der zum Suchen von Rollen innerhalb der `roleBase` verwendet wird. Der definierte Name des Benutzers, der mit `userSearchMatching` übereinstimmt, wird mit dem Platzhalter `{0}` im Suchfilter ersetzt. Der Benutzername des Kunden wird anstelle des `{1}`-Platzhalters eingesetzt. Wenn Rolleneinträge in Ihrem Verzeichnis beispielsweise ein Attribut mit dem Namen `member` enthalten, das die Benutzernamen für alle Benutzer in dieser Rolle enthält, können Sie den folgenden Suchfilter bereitstellen: `(member:=uid={1})`.

In der folgenden Abbildung wird hervorgehoben, wo diese Details angegeben werden sollen.



Im Abschnitt Optionale Einstellungen können Sie die folgenden optionalen Informationen angeben:

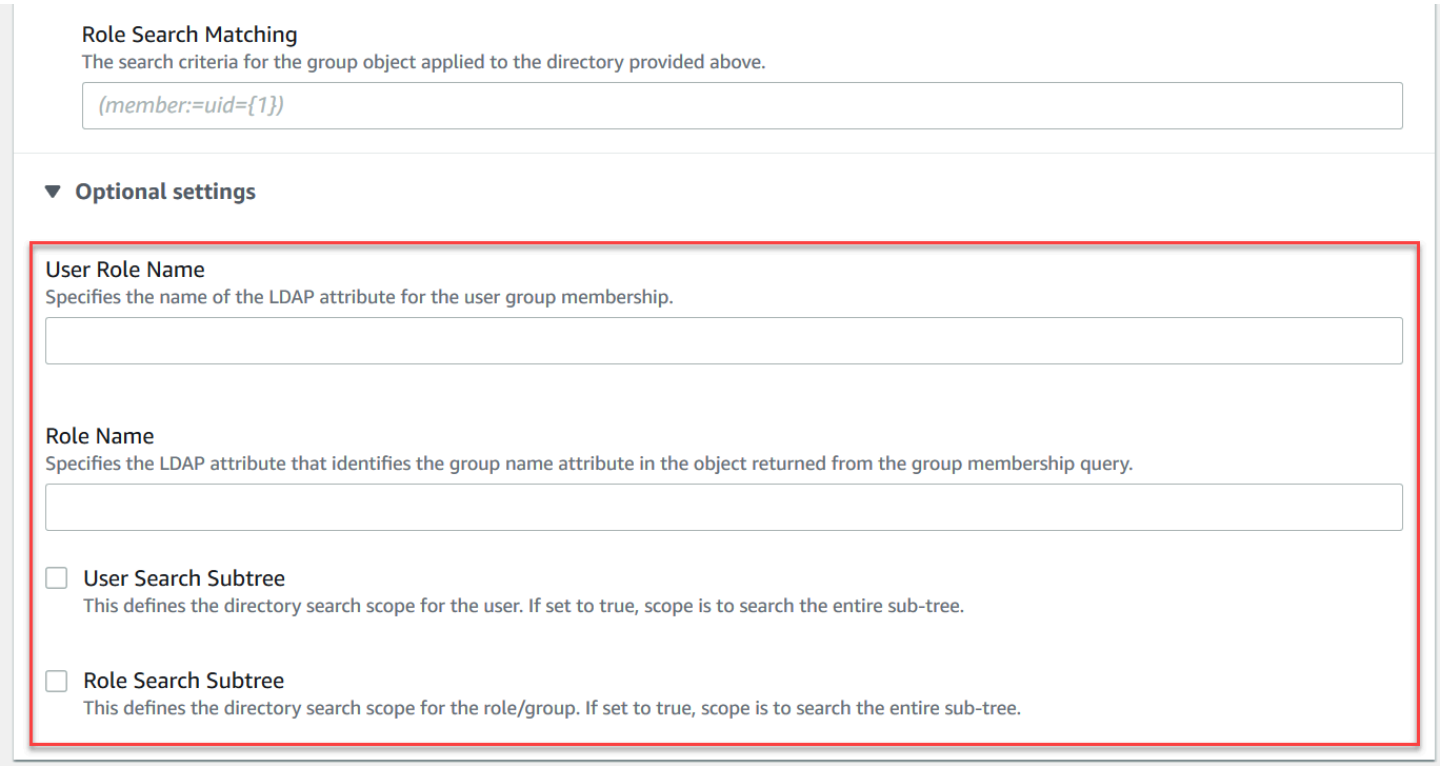
- Benutzerrollen-Name Der Name des LDAP-Attributs im Verzeichniseintrag des Benutzers für die Gruppenmitgliedschaft des Benutzers. In einigen Fällen können Benutzerrollen durch den Wert eines Attributs im Verzeichniseintrag des Benutzers identifiziert werden. Mit der `userRoleName`-Option können Sie den Namen dieses Attributs angeben. Betrachten wir beispielsweise den folgenden Benutzereintrag:

```
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
```

Um für das obige Beispiel den richtigen `userRoleName` bereitzustellen, würden Sie das `memberOf`-Attribut angeben. Wenn die Authentifizierung erfolgreich ist, wird dem Benutzer die `role1`-Rolle zugewiesen.

- Rollenname Das Gruppennamen-Attribut in einem Rolleneintrag, dessen Wert der Name dieser Rolle ist. Sie können beispielsweise `cn` für einen allgemeinen Namen eines Gruppeneintrags angeben. Wenn die Authentifizierung erfolgreich ist, wird dem Benutzer der Wert des Attributs `cn` für jeden Rolleneintrag zugewiesen, bei dem er Mitglied ist.
- Der Teilbaum Benutzersuche Definiert den Bereich für die LDAP-Benutzersuchabfrage. Wenn true, wird der Bereich so eingestellt, dass der gesamte Teilbaum unter dem Knoten durchsucht wird, der durch `userBase` definiert ist.
- Der Teilbaum Rollensuche Definiert den Bereich für die LDAP-Rollensuchabfrage. Wenn true, wird der Bereich so eingestellt, dass der gesamte Teilbaum unter dem Knoten durchsucht wird, der durch `roleBase` definiert wird.

In der folgenden Abbildung wird hervorgehoben, wo diese optionalen Einstellungen festgelegt werden sollen.



Funktionsweise der LDAP-Integration

Sie können sich die Integration in zwei Hauptkategorien vorstellen: die Struktur für die Authentifizierung und die Struktur für die Autorisierung.

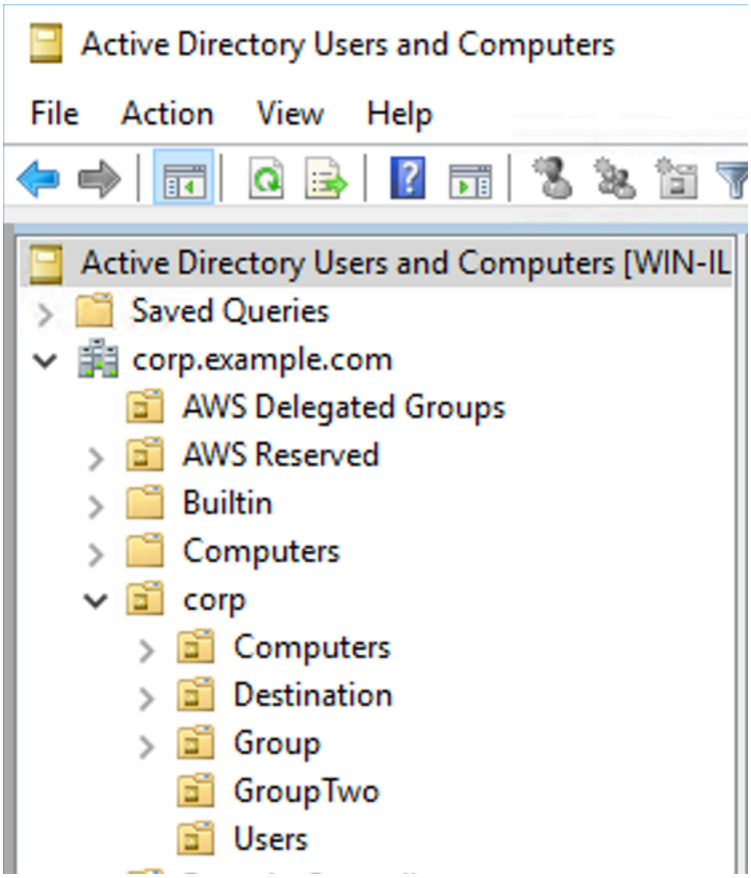
Authentifizierung

Für die Authentifizierung müssen Clientanmeldeinformationen gültig sein. Diese Anmeldeinformationen werden für Benutzer in der Benutzerbasis auf dem LDAP-Server validiert.

Die Benutzerbasis, die dem ActiveMQ-Broker bereitgestellt wird, muss auf den Knoten im DIT verweisen, auf dem Benutzer auf dem LDAP-Server gespeichert sind. Wenn Sie beispielsweise AWS Managed Microsoft AD verwenden, und Sie die Domänenkomponenten `corp`, `example`, und `com` haben, und innerhalb diesen die Organisationseinheiten `corp` und `Users`, würden Sie folgendes als Benutzerbasis verwenden:

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

Der ActiveMQ-Broker würde an diesem Speicherort im DIT nach Benutzern suchen, um Client-Verbindungsanforderungen an den Broker zu authentifizieren.



Da der ActiveMQ-Quellcode den Attributnamen für Benutzer zu `uid` festcodiert, müssen Sie sicherstellen, dass für jeden Benutzer dieses Attribut festgelegt ist. Der Einfachheit halber können Sie den Verbindungsbenutzernamen des Benutzers verwenden. Weitere Informationen finden Sie im ativemq-Quellcode und Konfigurieren von ID-Zuweisungen in Active-Directory-Benutzer und -Computer für Windows Server 2016 (und nachfolgenden) Versionen.

Um den ActiveMQ-Konsolenzugriff für bestimmte Benutzer zu aktivieren, stellen Sie sicher, dass sie zur `amazonmq-console-admins`-Gruppe gehören.

Autorisierung

Für die Autorisierung werden Berechtigungen Suchbasen in der Broker-Konfiguration angegeben. Die Autorisierung erfolgt pro Ziel (oder Platzhalter, Zielsatz) über das `cachedLdapAuthorizationMap`-Element, das sich in der `activemq.xml`-Konfigurationsdatei des Brokers befindet. Weitere Informationen finden Sie unter Zwischengespeichertes LDAP-Autorisierungsmodul.

> **Note**
>
> Um das `cachedLDAPAuthorizationMap`-Element in der `activemq.xml`-Konfigurationsdatei Ihres Brokers verwenden zu können, müssen Sie die Option LDAP Authentication and Authorization (LDAP-Authentifizierung und -Autorisierung) wählen, wenn Sie eine Konfiguration über den AWS Management Console erstellen, oder die `authenticationStrategy`-Eigenschaft auf `LDAP` setzen, wenn Sie eine neue Konfiguration über die Amazon-MQ-API erstellen.

Sie müssen die folgenden drei Attribute im Rahmen des `cachedLDAPAuthorizationMap`-Elements bereitstellen:

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

> **Important**
>
> Um zu verhindern, dass vertrauliche Informationen direkt in der Konfigurationsdatei des Brokers platziert werden, blockiert Amazon MQ die folgenden Attribute in`cachedLdapAuthorizationMap`:
>
> - `connectionURL`
> - `connectionUsername`
> - `connectionPassword`
>
> Wenn Sie einen Broker erstellen, ersetzt Amazon MQ die Werte, die Sie über die AWS Management Console, oder in der `ldapServerMetadata`-Eigenschaft Ihrer API-Anfrage für die obigen Attribute angeben.

Das folgende Beispiel illustriert die Verwendung von Verschiebungen.

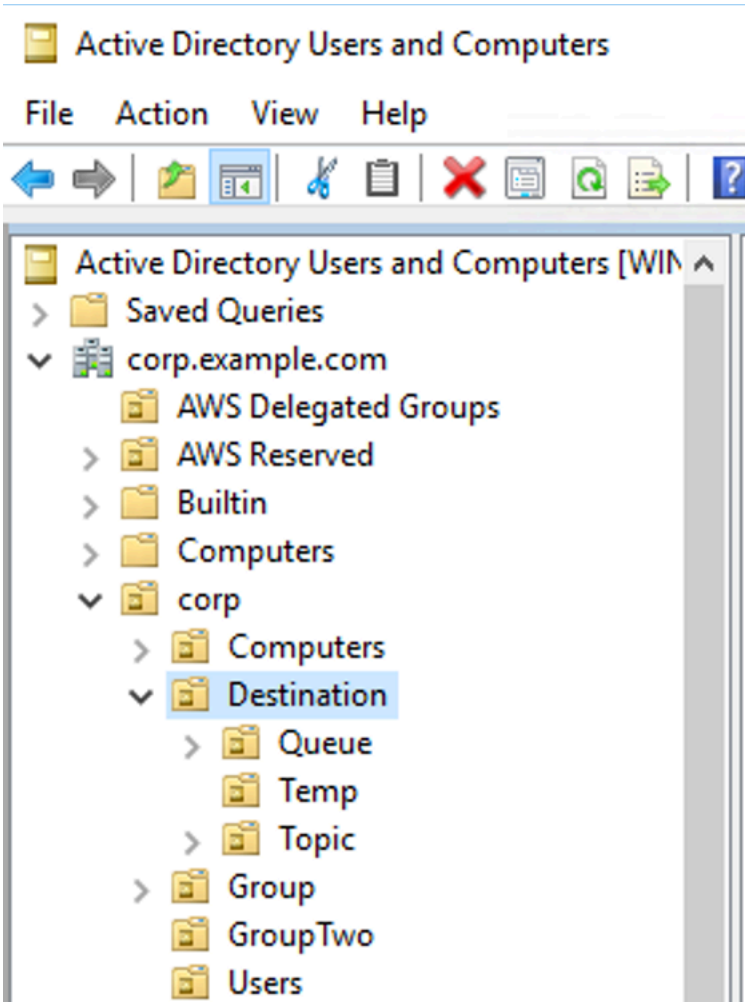
```
<authorizationPlugin>
    <map>
```

```
        <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>
```

Diese Werte geben die Speicherorte innerhalb des DIT an, an denen Berechtigungen für jeden Zieltyp angegeben werden. Also für das obige Beispiel mit AWS Managed Microsoft AD, wobei die gleichen Domänenkomponenten von `corp`, `example`, und `com` verwendet werden, geben Sie eine Organisationseinheit mit dem Namen `destination` an, um alle Zieltypen zu enthalten. Innerhalb dieser Organisationseinheit würden Sie jeweils eine für die Ziele `queues`, `topics` und `temp` erstellen.

Dies würde bedeuten, dass Ihre Warteschlangen-Suchbasis, die Autorisierungsinformationen für Ziele vom Typ Warteschlange bereitstellt, den folgenden Speicherort in Ihrem DIT hat:

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



Ebenso würden Berechtigungsregeln für Themen und temporäre Ziele auf der gleichen Ebene im DIT liegen:

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

Innerhalb der Organisationseinheit für jeden Zieltyp (Warteschlange, Thema, Temp) kann entweder ein Platzhalter oder ein bestimmter Zielname angegeben werden. Um beispielsweise eine Autorisierungsregel für alle Warteschlangen bereitzustellen, die mit dem Präfix DEMO.EVENTS.$ beginnen, können Sie die folgende Organisationseinheit erstellen:

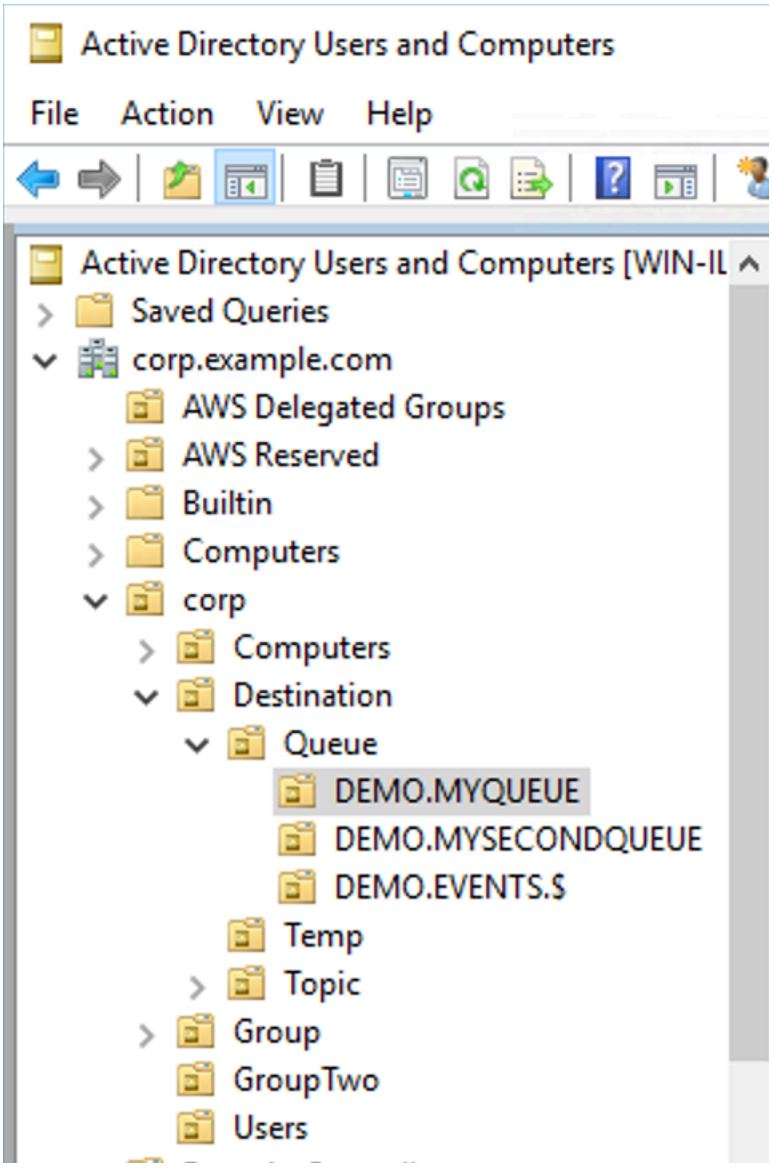
```
OU=DEMO.EVENTS.$,OU=Queue,OU=Destination,OU=corp,DC=corp,DC=example,DC=com
```

> **Note**
>
> Die `DEMO.EVENTS.$`-Organisationseinheit befindet sich innerhalb der `Queue`-Organisationseinheit.

Weitere Informationen zu Platzhaltern in ActiveMQ finden Sie unter Platzhalter

Um Autorisierungsregeln für bestimmte Warteschlangen wie DEMO.MYQUEUE bereitzustellen, geben Sie Folgendes an:

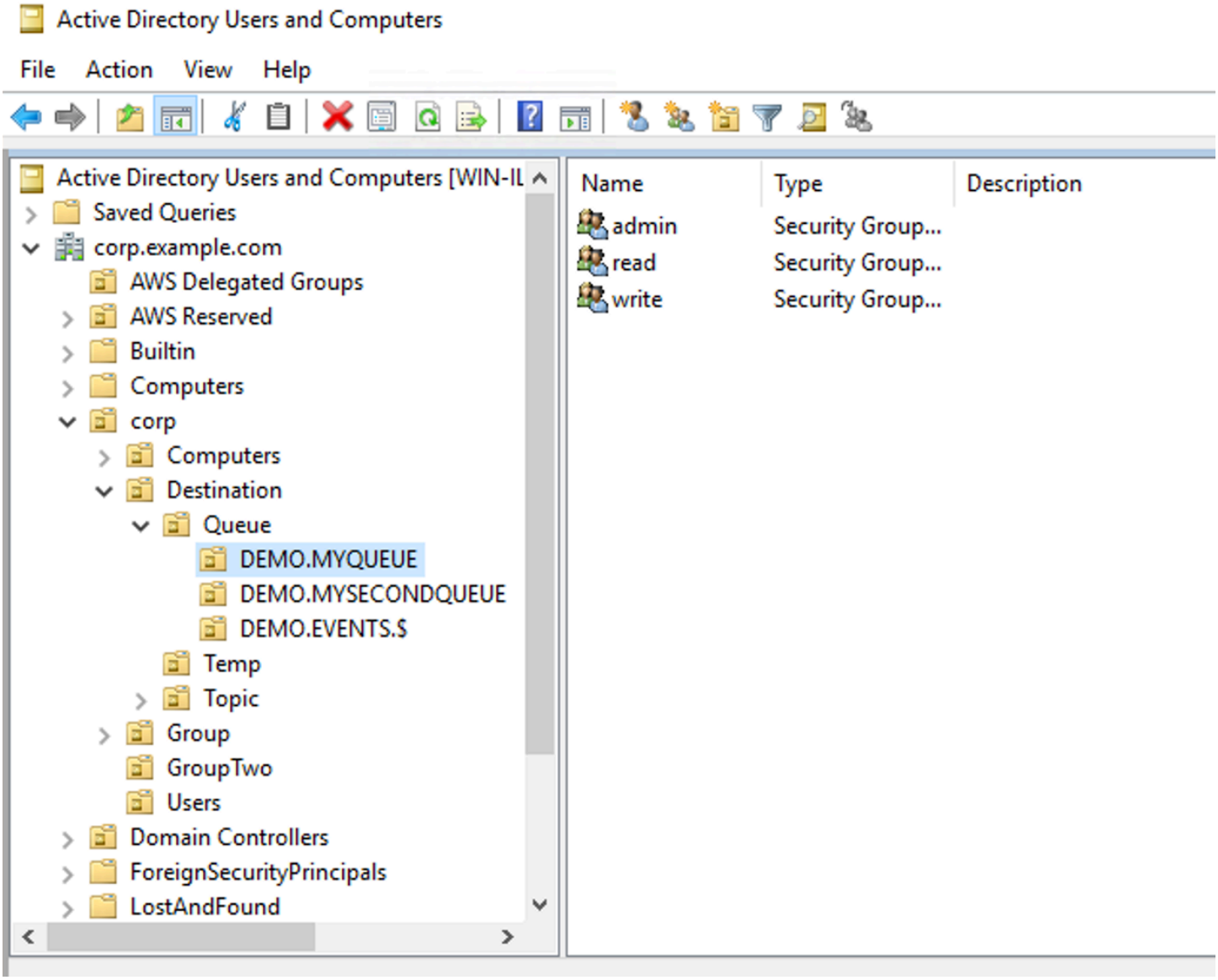
```
OU=DEMO.MYQUEUE,OU=Queue,OU=Destination,OU=corp,DC=corp,DC=example,DC=com
```



Sicherheitsgruppen

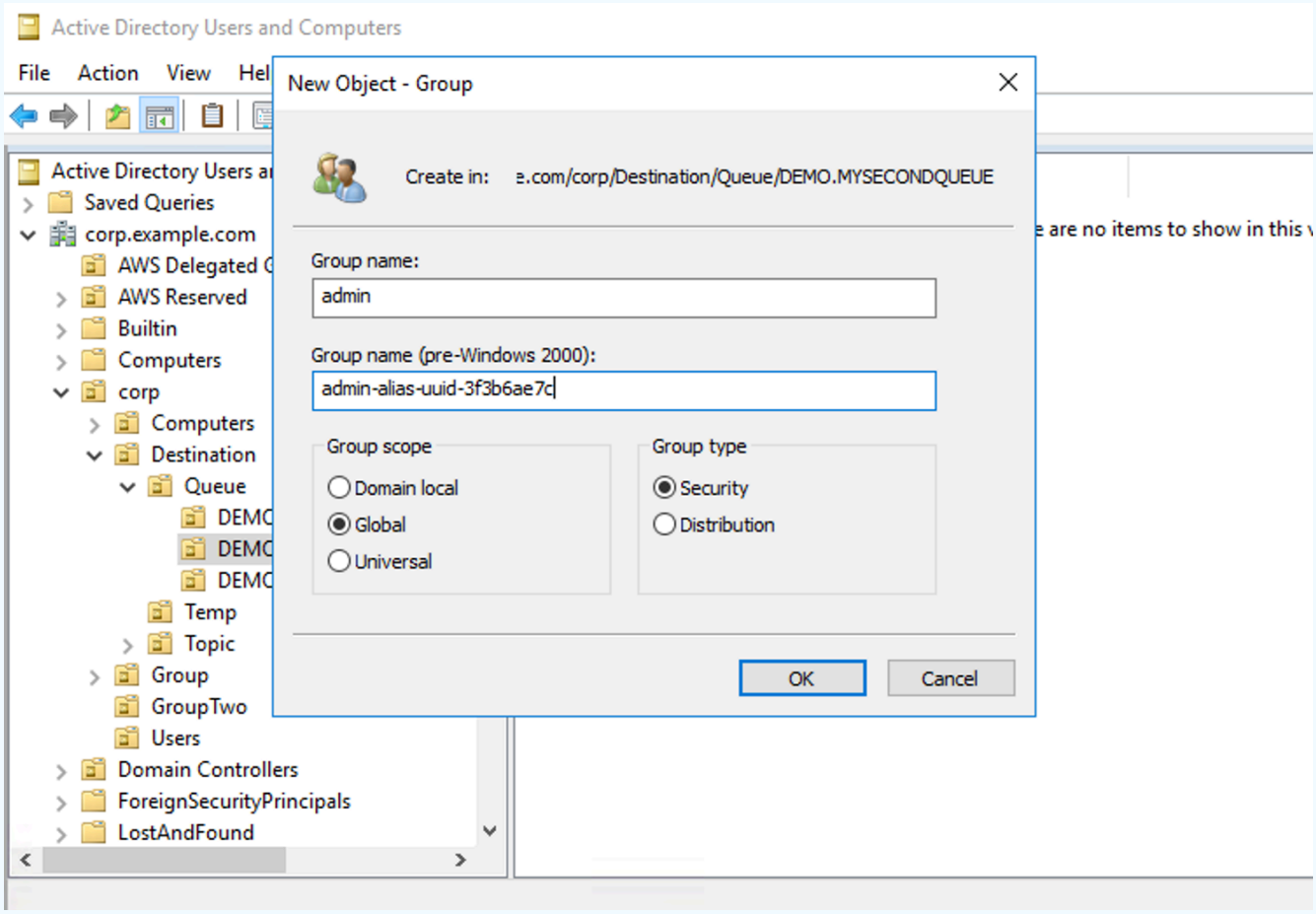
Innerhalb jeder Organisationseinheit, die ein Ziel oder einen Platzhalter darstellt, müssen Sie drei Sicherheitsgruppen erstellen. Wie bei allen Berechtigungen in ActiveMQ handelt es sich hierbei um Lese-/Schreib-/Administratorberechtigungen. Weitere Informationen zu den Funktionen der einzelnen Berechtigungen eines Benutzers finden Sie unter Sicherheit in der ActiveMQ-Dokumentation.

Sie müssen diese Sicherheitsgruppen `read`, `write` und `admin` benennen. Innerhalb jeder dieser Sicherheitsgruppen können Sie Benutzer oder Gruppen hinzufügen, die dann über die Berechtigung zum Ausführen der zugehörigen Aktionen verfügen. Sie benötigen diese Sicherheitsgruppen für jede Platzhalterzielgruppe oder jedes einzelne Ziel.



Note

Wenn Sie die Admin-Gruppe erstellen, entsteht ein Konflikt mit dem Gruppennamen. Dieser Konflikt tritt auf, weil die Legacy-Regeln vor Windows 2000 nicht zulassen, dass Gruppen denselben Namen verwenden, selbst wenn sich die Gruppen an unterschiedlichen Speicherorten des DIT befinden. Der Wert in dem Dialogfeld pre-Windows 2000 hat keine Auswirkungen auf die Einrichtung, muss jedoch global eindeutig sein. Um diesen Konflikt zu vermeiden, können Sie ein `uuid`-Suffix jeder `admin`-Gruppe anknüpfen.



Hinzufügen eines Benutzers zur `admin`-Sicherheitsgruppe für ein bestimmtes Ziel ermöglicht es dem Benutzer, dieses Thema zu erstellen und zu löschen. Sie zur `read`-Sicherheitsgruppe hinzuzufügen ermöglicht es ihnen, vom Ziel zu lesen und sie der `write`-Gruppe hinzuzufügen ermöglicht es ihnen, an das Ziel zu schreiben.

Zusätzlich zum Hinzufügen einzelner Benutzer zu Sicherheitsgruppen-Berechtigungen können Sie auch ganze Gruppen hinzufügen. Da ActiveMQ jedoch wieder Attributnamen für Gruppen

festcodiert, müssen Sie sicherstellen, dass die Gruppe, die Sie hinzufügen möchten, die Objektklasse `groupOfNames` hat, wie im activemq-Quellcode beschrieben.

Führen Sie dazu den gleichen Prozess aus wie bei der `uid` für Benutzer. Siehe Konfigurieren von ID-Zuweisungen in Active-Directory-Benutzern und Computer für Windows Server 2016 (und nachfolgenden) Versionen.

Einen ActiveMQ-Broker-Benutzer erstellen

Ein ActiveMQBenutzerist eine Person oder eine Anwendung, die auf die Warteschlangen und Themen eines ActiveMQ -Brokers zugreifen kann. Sie können Benutzer so konfigurieren, dass sie bestimmte Berechtigungen haben. Beispielsweise können Sie einigen Benutzern erlauben, auf dieActiveMQ-Webkonsole zuzugreifen.

Eine Gruppeist ein semantisches Label. Sie können einem Benutzer eine Gruppe zuweisen und Berechtigungen für Gruppen zum Senden, Empfangen von und Verwalten bestimmter Warteschlangen und Themen konfigurieren.

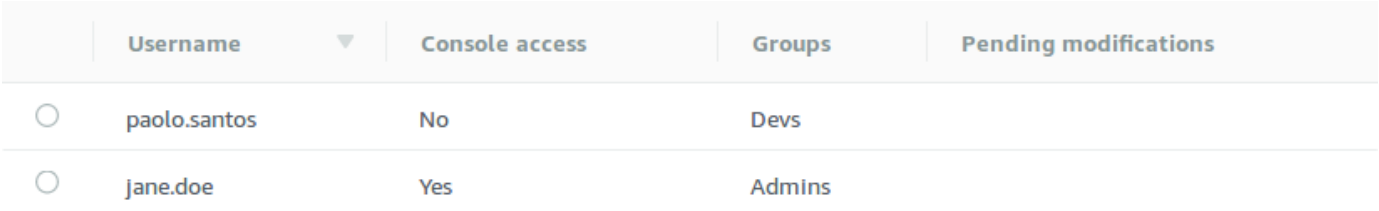
> **Note**
>
> Sie können Gruppen nicht unabhängig von Benutzern konfigurieren. Eine Gruppenbezeichnung wird erstellt, wenn Sie mindestens einen Benutzer hinzufügen und gelöscht, wenn Sie alle Benutzer daraus entfernen.

Die folgenden Beispiele zeigen, wie Sie Amazon MQ-Broker-Benutzer mithilfe der AWS Management Console erstellen, bearbeiten und löschen können.

Neuen ActiveMQ-Broker-Benutzer erstellen

1. Melden Sie sich bei der Amazon-MQ-Konsole an.
2. Wählen Sie in der Brokerliste den Namen Ihres Brokers aus (z. B. MyBroker) und wählen Sie dann Details anzeigen aus.

   Auf der ***MyBroker***Auf der Seite werden im Bereich Benutzer alle Benutzer dieses Brokers aufgelistet.



3. Wählen Sie Create user (Benutzer erstellen) aus.
4. Geben Sie in das Dialogfeld Create user (Benutzer erstellen) einen Benutzernamen und ein Kennwort ein.
5. (Optional) Geben Sie durch Kommas voneinander getrennt die Namen der Gruppen ein, denen der Benutzer angehört (z. B.: `Devs, Admins`).
6. (Optional) Um dem Benutzer zu ermöglichen, auf die ActiveMQ-Webkonsole zuzugreifen, wählen Sie ActiveMQ Web Console.
7. Wählen Sie Create user (Benutzer erstellen) aus.

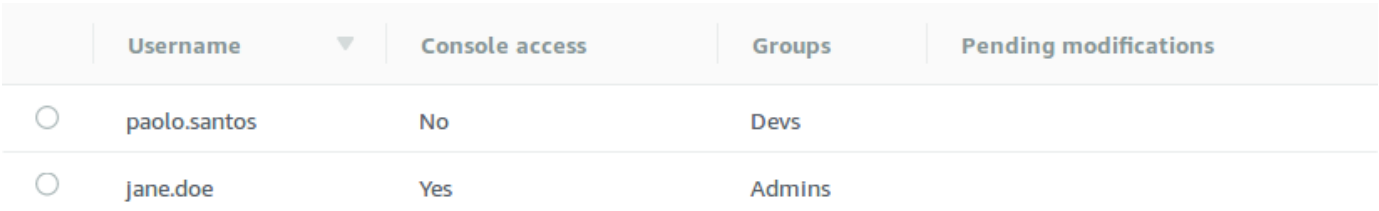
> **Important**
>
> Das Vornehmen von Änderungen an einem Benutzer wendet nicht sofort die Änderungen auf den Benutzer an. Um Ihre Änderungen zu übernehmen, müssen Sie auf den nächsten Wartungszeitraum warten oder den Broker neu starten.

Einen ActiveMQ-Broker-Benutzer bearbeiten

Gehen Sie wie folgt vor, um einen vorhandenen Benutzer zu bearbeiten:

1. Melden Sie sich bei der Amazon-MQ-Konsole an.
2. Wählen Sie in der Brokerliste den Namen Ihres Brokers aus (z. B. MyBroker) und wählen Sie dann Details anzeigen aus.

   Auf der ***MyBroker***Auf der Seite werden im Bereich Benutzer alle Benutzer dieses Brokers aufgelistet.



3. Wählen Sie Ihre Anmeldeinformationen und dann Bearbeiten aus.

   Das Dialogfeld Edit user (Benutzer bearbeiten) wird angezeigt.

4. (Optional) Geben Sie ein neues Kennwort ein.
5. (Optional) Fügen Sie die durch Kommas voneinander getrennten Namen der Gruppen, denen der Benutzer angehört, hinzu oder entfernen Sie sie (z. B.: `Managers, Admins`).
6. (Optional) Um dem Benutzer zu ermöglichen, auf die ActiveMQ-Webkonsole zuzugreifen, wählen Sie ActiveMQ Web Console.
7. Um die Änderungen am Benutzer zu speichern, wählen Sie Done (Fertig) aus.

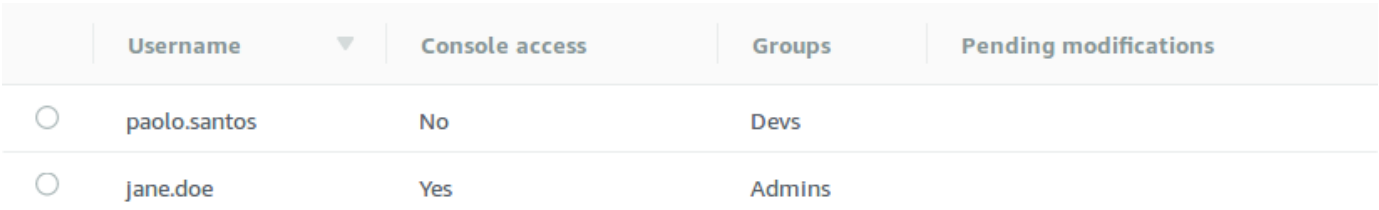
> **Important**
>
> Das Vornehmen von Änderungen an einem Benutzer wendet nicht sofort die Änderungen auf den Benutzer an. Um Ihre Änderungen zu übernehmen, müssen Sie auf den nächsten Wartungszeitraum warten oder den Broker neu starten.

Löschen Sie einen ActiveMQ-Broker-Benutzer

Wenn Sie einen Benutzer nicht mehr benötigen, können Sie ihn löschen.

1. Melden Sie sich bei der Amazon-MQ-Konsole an.
2. Wählen Sie in der Brokerliste den Namen Ihres Brokers aus (z. B. MyBroker) und wählen Sie dann Details anzeigen aus.

   Auf der ***MyBroker***Auf der Seite werden im Bereich Benutzer alle Benutzer dieses Brokers aufgelistet.



3. Wählen Sie Ihre Anmeldedaten aus (z. B. ***MyUser***) und wählen Sie dann Löschen.
4. Um das Löschen des Benutzers zu bestätigen, klicken Sie auf Löschen ***MyUser***? Wählen Sie im Dialogfeld Löschen.

> **Important**
>
> Das Vornehmen von Änderungen an einem Benutzer wendet nicht sofort die Änderungen auf den Benutzer an. Um Ihre Änderungen zu übernehmen, müssen Sie auf den nächsten Wartungszeitraum warten oder den Broker neu starten.

Arbeitsbeispiele für die Verwendung von Java Message Service (JMS) mit ActiveMQ

Die folgenden Beispiele zeigen, wie Sie programmgesteuert mit ActiveMQ arbeiten können:

- Der OpenWire Java-Beispielcode stellt eine Verbindung zu einem Broker her, erstellt eine Warteschlange und sendet und empfängt eine Nachricht. Eine detaillierte Aufschlüsselung und Erläuterung finden Sie unter Connecting a Java application to your broker.
- Der MQTT Java-Beispielcode stellt eine Verbindung zu einem Broker her, erstellt ein Thema und veröffentlicht und empfängt eine Nachricht.
- Der WSS Java-Beispielcode STOMP + stellt eine Verbindung zu einem Broker her, erstellt eine Warteschlange und veröffentlicht und empfängt eine Nachricht.

Voraussetzungen

VPCAttribute aktivieren

Um sicherzustellen, dass Ihr Broker in Ihrem erreichbar istVPC, müssen Sie die `enableDnsSupport` VPC Attribute `enableDnsHostnames` und aktivieren. Weitere Informationen finden Sie unter DNSSupport VPC in Ihrem VPC Amazon-Benutzerhandbuch.

Eingehende Verbindungen aktivieren

Um programmgesteuert mit Amazon MQ arbeiten zu können, müssen Sie eingehende Verbindungen verwenden.

1. Melden Sie sich bei der Amazon-MQ-Konsole an.
2. Wählen Sie aus der Brokerliste den Namen Ihres Brokers aus (z. B.). MyBroker
3. Auf der ***MyBroker***Notieren Sie sich im Abschnitt Verbindungen die Adressen und Ports der Webkonsole URL und der Wire-Level-Protokolle des Brokers.

4. Wählen Sie im Abschnitt Details unter Sicherheit und Netzwerk den Namen Ihrer Sicherheitsgruppe oder

   .

   Die Seite „Sicherheitsgruppen“ des EC2 Dashboards wird angezeigt.
5. Wählen Sie in der Liste der Sicherheitsgruppen Ihre Sicherheitsgruppe.
6. Klicken Sie unten auf der Seite auf Inbound (Eingehend) und anschließend auf Edit (Bearbeiten).
7. Fügen Sie im Dialogfeld „Regeln für eingehenden Datenverkehr bearbeiten“ eine Regel für jeden URL oder Endpunkt hinzu, auf den Sie öffentlich zugreifen möchten (das folgende Beispiel zeigt, wie Sie dies für eine Broker-Webkonsole tun können).

   a. Klicken Sie auf Add Rule (Regel hinzufügen).
   b. Wählen Sie für Typ die Option Benutzerdefiniert TCP aus.
   c. Für Port-Bereich, geben Sie den Port der Webkonsole ein (`8162`).
   d. Für Source (Quelle), lassen Sie Custom (Benutzerdefiniert) ausgewählt, und geben Sie dann die IP-Adresse des Systems ein, auf das auf die Webkonsole zugegriffen werden soll (z. B. `192.0.2.1`) enthalten.
   e. Wählen Sie Save.

      Ihr Broker kann nun eingehende Verbindungen akzeptieren.

Java-Abhängigkeiten hinzufügen

OpenWire

Fügen Sie dem Pfad für Ihre Java-Build-Klasse die Pakete `activemq-client.jar` und `activemq-pool.jar`hinzu. Das folgende Beispiel zeigt diese Abhängigkeiten in der `pom.xml`-Datei eines Maven-Projekts.

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

```
        <version>5.15.16</version>
    </dependency>
</dependencies>
```

Weitere Informationen über `activemq-client.jar` finden Sie unter Ursprüngliche Konfiguration in der Apache ActiveMQ-Dokumentation.

MQTT

Fügen Sie dem Pfad für Ihre Java-Klasse das `org.eclipse.paho.client.mqttv3.jar`-Pakete hinzu. Das folgende Beispiel zeigt diese Abhängigkeit in der `pom.xml`-Datei eines Maven-Projekts.

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

Weitere Informationen zu `org.eclipse.paho.client.mqttv3.jar` finden Sie unter Eclipse Paho-Java-Client.

STOMP+WSS

Fügen Sie die folgenden Pakete zu Ihrem Java-Klassenpfad hinzu:

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

Das folgende Beispiel zeigt diese Abhängigkeiten in der `pom.xml`-Datei eines Maven-Projekts.

```
<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>
```

Weitere Informationen finden Sie unter STOMPSupport in der Spring Framework-Dokumentation.

Ein mazonMQExample .java

> **Important**
>
> Im folgenden Beispielcode laufen Hersteller und Verbraucher in einem einzigen Thread. Stellen Sie für Produktionssysteme (oder zum Testen des Failovers von Broker-Instances) sicher, dass Ihre Produzenten und Verbraucher auf separaten Hosts oder Threads ausgeführt werden.

OpenWire

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

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

                    import javax.jms.*;

                    public class AmazonMQExample {

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

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

                        sendMessage(pooledConnectionFactory);
                        receiveMessage(connectionFactory);

                        pooledConnectionFactory.stop();
```

```
                    }

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

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

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

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

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

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

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

                    private static void
                    receiveMessage(ActiveMQConnectionFactory connectionFactory)
throws JMSException {
                        // Establish a connection for the consumer.
                        // Note: Consumers should not use PooledConnectionFactory.
```

```
                        final Connection consumerConnection =
connectionFactory.createConnection();
                        consumerConnection.start();

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

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

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

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

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

                        // Clean up the consumer.
                        consumer.close();
                        consumerSession.close();
                        consumerConnection.close();
                    }

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

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

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

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

MQTT

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

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

                    public class AmazonMQExampleMqtt implements MqttCallback {

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

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

                    private void run() throws MqttException, InterruptedException {

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

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

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

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

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

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

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

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

                    @Override
```

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

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

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

STOMP+WSS

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

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

```
                    import
 org.springframework.web.socket.messaging.WebSocketStompClient;

                    import java.lang.reflect.Type;

                    public class AmazonMQExampleStompWss {

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

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

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

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

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

                        final WebSocketHttpHeaders headers = new
 WebSocketHttpHeaders();

                        // Create headers with authentication parameters.
```

```
                        final StompHeaders head = new StompHeaders();
                        head.add(StompHeaders.LOGIN, ACTIVE_MQ_USERNAME);
                        head.add(StompHeaders.PASSCODE, ACTIVE_MQ_PASSWORD);

                        final StompSessionHandler sessionHandler = new
MySessionHandler();

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

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

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

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

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

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

```
}
```

Verwalten von Amazon MQ für ActiveMQ Engine-Versionen

Apache ActiveMQ organisiert Versionsnummern gemäß der semantischen Versionsspezifikation als`X.Y.Z`. `X`Bezeichnet in Amazon MQ für ActiveMQ-Implementierungen die Hauptversion, `Y` steht für die Nebenversion und gibt die Patch-Versionsnummer an`Z`. Amazon MQ betrachtet eine Versionsänderung als Hauptversionsänderung, wenn sich die Hauptversionsnummern ändern. Beispielsweise wird ein Upgrade von Version 5.17 auf 6.0 als Hauptversions-Upgrade betrachtet. Eine Versionsänderung gilt als geringfügig, wenn sich nur die Versionsnummer der Nebenversion oder des Patches ändert. Zum Beispiel ein Upgrade von Version 5. 17 auf 5. 18 wird als geringfügiges Versionsupgrade betrachtet.

Amazon MQ for ActiveMQ empfiehlt allen Brokern, die neueste unterstützte Nebenversion zu verwenden. Anweisungen zum Upgrade Ihrer Broker-Engine-Version finden Sie unter Upgrade einer Amazon MQ-Broker-Engine-Version.

Unterstützte Engine-Versionen auf Amazon MQ für ActiveMQ

Der Support-Kalender für die Amazon MQ MQ-Version gibt an, wann der Support für eine Broker-Engine-Version endet. Wenn für eine Version der Support ausläuft, aktualisiert Amazon MQ alle Broker dieser Version automatisch auf die nächste unterstützte Version. Dieses Upgrade findet während der geplanten Wartungsfenster Ihres Brokers innerhalb von 45 Tagen nach dem end-of-support Datum statt.

Amazon MQ informiert Sie mindestens 90 Tage im Voraus, bevor der Support für eine Version endet. Wir empfehlen, Ihren Broker vor diesem end-of-support Datum zu aktualisieren, um Störungen zu vermeiden. Darüber hinaus können Sie innerhalb von 30 Tagen nach Ablauf des Supports keine neuen Broker für Versionen erstellen, für die das Ende des Supports geplant ist.

Apache ActiveMQ-Version	Ende des Supports bei Amazon MQ
ActiveMQ 5.18 (empfohlen)	
ActiveMQ 5.17	
ActiveMQ 5.16	15. November 2024

Apache ActiveMQ-Version	Ende des Supports bei Amazon MQ
ActiveMQ 5.16	16. September 2024

Wenn Sie einen neuen Amazon MQ für ActiveMQ Broker erstellen, können Sie jede unterstützte ActiveMQ Engine-Version angeben. Wenn Sie bei der Erstellung eines Brokers keine Engine-Versionsnummer angeben, verwendet Amazon MQ automatisch standardmäßig die neueste Engine-Versionsnummer.

Upgrades der Engine-Version

Sie können Ihren Broker jederzeit manuell auf die nächste unterstützte Haupt- oder Nebenversion aktualisieren. Wenn Sie automatische Upgrades für Nebenversionen aktivieren, aktualisiert Amazon MQ Ihren Broker während des Wartungsfensters auf die neueste unterstützte Patch-Version.

Weitere Informationen zur manuellen Aktualisierung Ihres Brokers finden Sie unterthe section called “Upgrade der Engine-Version”.

Unterstützte Engine-Versionen auflisten

Mithilfe des `describe-broker-instance-options` AWS CLI Befehls können Sie alle unterstützten Neben- und Hauptversionen der Engine auflisten.

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

Um die Ergebnisse nach Engine und Instance-Typ zu filtern, verwenden Sie die `--engine-type`- und `--host-instance-type`-Optionen wie im Folgenden gezeigt.

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

Um beispielsweise die Ergebnisse nach ActiveMQ und `mq.m5.large` Instanztyp zu filtern, ersetzen Sie *`engine-type`* mit und `ACTIVEMQ` *`instance-type`* mit`mq.m5.large`.

Amazon MQ für ActiveMQ-Speichertypen

Amazon MQ for ActiveMQ unterstützt Amazon Elastic File System (EFS) und Amazon Elastic Block Store (). EBS Standardmäßig verwenden ActiveMQ-Broker Amazon als EFS Broker-Speicher.

Verwenden Sie Amazon, um die Vorteile der hohen Haltbarkeit und Replikation über mehrere Availability Zones hinweg zu nutzenEFS. Verwenden Sie Amazon, um von niedriger Latenz und hohem Durchsatz zu profitierenEBS.

> **Important**
>
> - Sie können Amazon EBS nur mit der `mq.m5` Broker-Instance-Typfamilie verwenden.
> - Obwohl Sie den Broker-Instance-Typ ändern können, ist es nicht möglich, den Speichertyp des Brokers zu ändern, nachdem Sie den Broker erstellt haben.
> - Amazon EBS repliziert Daten innerhalb einer einzigen Availability Zone und unterstützt den ActiveMQ-Aktiv-/Standby-Bereitstellungsmodus nicht.

Unterschiede zwischen Speichertypen

Die folgende Tabelle bietet einen kurzen Überblick über die Unterschiede zwischen In-Memory-EFS, Amazon- und EBS Amazon-Speichertypen für ActiveMQ-Broker.

Speichertyp	Persistenz	Beispielanwendungsfall	Ungefähre maximale Anzahl von Nachrichten, die pro Produzent pro Sekunde (1-KB-Nachricht) in die Warteschlange gestellt werden	Replikation
In-Memory	Nicht persistent	• Aktienkurse • Aktualisierungen von Standortdaten • Häufig geänderte Daten	5,000	None

Speichertyp	Persistenz	Beispielanwendungsfall	Ungefähre maximale Anzahl von Nachrichten, die pro Produzent pro Sekunde (1-KB-Nachricht) in die Warteschlange gestellt werden	Replikation
Amazon EBS	Persistent	• Umfangreiche Textmengen • Antragsbearbeitung	500	Mehrere Kopien innerhalb einer einzigen Availability Zone (AZ)
Amazon EFS	Persistent	Finanztransaktionen	80	Mehrere Kopien über mehrere AZs

Der In-Memory-Nachrichtenspeicher bietet die niedrigste Latenz und den höchsten Durchsatz. Nachrichten gehen jedoch während der Instance-Ersetzung oder des Neustarts des Brokers verloren.

Amazon EFS ist so konzipiert, dass es äußerst robust ist und über mehrere repliziert wird, AZs um den Verlust von Daten zu verhindern, der durch den Ausfall einer einzelnen Komponente oder durch ein Problem entsteht, das die Verfügbarkeit einer AZ beeinträchtigt. Amazon EBS ist für den Durchsatz optimiert und wird auf mehreren Servern innerhalb einer einzigen AZ repliziert.

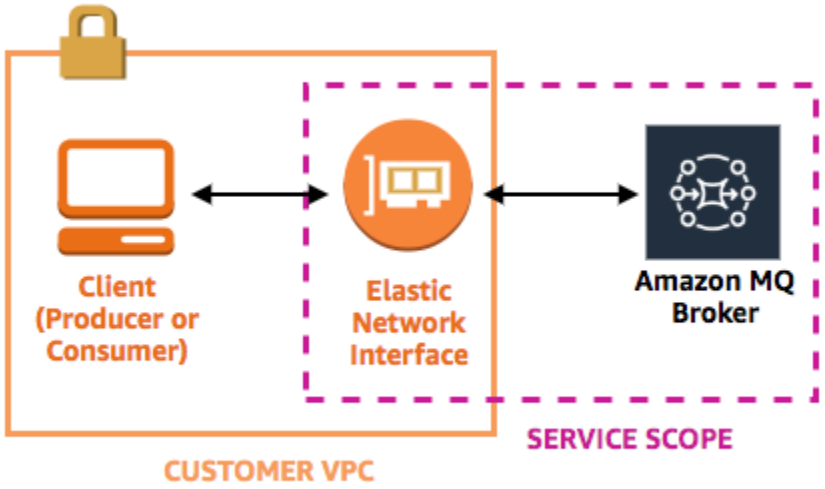
Best Practices für Amazon MQ für ActiveMQ

In diesem Abschnitt finden Sie schnell Empfehlungen für die Maximierung der Leistung und die Minimierung der Durchsatzkosten bei der Arbeit mit ActiveMQ brokers auf Amazon MQ.

Verändern oder löschen Sie auf keinen Fall die Amazon MQ Elastic Network-Schnittstelle

Wenn Sie zum ersten Mal einen Amazon MQ-Broker erstellen, stellt Amazon MQ eine elastic network interface in der Virtual Private Cloud (VPC) unter Ihrem Konto bereit und benötigt daher eine Reihe von EC2 Berechtigungen. Die Netzwerkschnittstelle gestattet Ihrem Client (Erzeuger oder Verbraucher), mit dem Amazon MQ-Broker zu kommunizieren. Es wird davon ausgegangen, dass die Netzwerkschnittstelle zum Serviceumfang von Amazon MQ gehört, obwohl sie Teil Ihres Kontos istVPC.

> **Warning**
>
> Sie dürfen diese Netzwerkschnittstelle nicht ändern oder löschen. Das Ändern oder Löschen der Netzwerkschnittstelle kann zu einem dauerhaften Verlust der Verbindung zwischen Ihnen VPC und Ihrem Broker führen.



Verwenden Sie immer Verbindungspools

In einem Szenario mit einem einzigen Produzenten und einem einzigen Konsumenten (z. B. das Erste Schritte: Einen ActiveMQ-Broker erstellen und eine Verbindung zu ihm herstellen-Tutorial) können Sie eine einzige `ActiveMQConnectionFactory`-Klasse für jeden Produzenten und Konsumenten verwenden. Beispielsweise:

```
// 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();
```

In realistischeren Szenarien mit mehreren Produzenten und Konsumenten hingegen kann es teuer und ineffizient sein, eine große Anzahl von Verbindungen für mehrere Produzenten zu generieren. In diesen Szenarien sollten Sie mehrere Produzentenanfragen mithilfe der `PooledConnectionFactory`-Klasse gruppieren. Beispielsweise:

> **Note**
>
> Die Nachrichtenkonsumenten sollten nie die `PooledConnectionFactory`-Klasse verwenden.

```
// 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();
```

Immer Failover-Transport verwenden, um Verbindungen zu mehreren Broker-Endpunkten einzurichten

Wenn Ihre Anwendung eine Verbindung zu mehreren Broker-Endpunkten einrichten muss – wenn Sie z. B. einen aktiven/Standby-Bereitstellungsmodus verwenden oder wenn Sie von einem lokalen Message Broker auf Amazon MQ migrieren –, verwenden Sie den Failover-Transport, um Ihren Konsumenten zu ermöglichen, eine Verbindung zu einem beliebigen dieser Endpunkte herzustellen. Beispielsweise:

```
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=true
```

Vermeiden Sie die Nachrichtenauswahl

Es ist möglich, mithilfe von JMSSelektoren Filter an Themenabonnements anzuhängen (um Nachrichten anhand ihres Inhalts an Verbraucher weiterzuleiten). Die Verwendung von JMS Selektoren füllt jedoch den Filterpuffer des Amazon MQ-Brokers und verhindert so, dass Nachrichten gefiltert werden.

Im Allgemeinen sollten Sie vermeiden, dass Konsumenten Nachrichten weiterleiten können, denn für eine optimale Entkopplung von Konsumenten und Produzenten sollte sowohl der Konsument als auch der Produzent flüchtig sein.

Virtuelle Ziele gegenüber dauerhaften Abonnements bevorzugen

Ein dauerhaftes Abonnement kann sicherstellen, dass der Konsument alle Nachrichten erhält, die zu einem Thema veröffentlicht werden, z. B. nach einer Verbindungswiederherstellung. Die Verwendung von dauerhaften Abonnements schließt jedoch auch die Verwendung konkurrierender Verbrauchern aus und kann bei einem großem Umfang zu Leistungsproblemen führen. Ziehen Sie stattdessen die Verwendung von virtuellen Zielen in Betracht.

Wenn Sie Amazon VPC Peering verwenden, vermeiden Sie den Client IPs in Reichweite CIDR **`10.0.0.0/16`**

Wenn Sie VPC Amazon-Peering zwischen der lokalen Infrastruktur und Ihrem Amazon MQ-Broker einrichten, dürfen Sie keine Client-Verbindungen mit IPs In-Range konfigurieren. CIDR `10.0.0.0/16`

Gleichzeitige Speicherung und Bereitstellung für Warteschlangen mit langsamen Konsumenten deaktivieren

Standardmäßig optimiert Amazon MQ für Warteschlangen mit schnellen Konsumenten:

- Konsumenten gelten als schnell, wenn sie in der Lage sind, mit der Rate der von Produzenten erstellten Nachrichten mitzuhalten.
- Konsumenten gelten als langsam, wenn sich in der Warteschlange ein Rückstand an nicht bestätigten Nachrichten aufbaut, was möglicherweise zu einer Verringerung des Durchsatzes des Produzenten führt.

Um Amazon MQ anzuweisen, für Warteschlange mit langsamen Konsumenten zu optimieren, legen Sie das Attribut `concurrentStoreAndDispatchQueues` auf `false` fest. Eine Beispielkonfiguration finden Sie unter `concurrentStoreAndDispatchQueues`.

Auswählen des richtigen Broker-Instance-Typs für den besten Durchsatz

Der Nachrichtendurchsatz eines Broker-Instance-Typs hängt von dem Anwendungsfall Ihrer Anwendung und den folgenden Faktoren ab:

- Verwendung von ActiveMQ im persistenten Modus
- Nachrichtengröße
- Anzahl an Produzenten und Konsumenten
- Anzahl an Zielen

Verstehen der Beziehung zwischen Nachrichtengröße, Latenz und Durchsatz

Je nach Ihrem Anwendungsfall lässt sich mit einem größeren Broker-Instance-Typ der Durchsatz möglicherweise nicht verbessern. Wenn ActiveMQ Nachrichten in einen Speicher mit hoher Beständigkeit schreibt, bestimmt die Größe Ihrer Nachrichten den begrenzenden Faktor Ihres Systems:

- Wenn Ihre Nachrichten kleiner als 100 KB sind, ist die Latenz des persistenten Speichers der begrenzende Faktor.
- Wenn Ihre Nachrichten größer als 100 KB sind, ist der Durchsatz des persistenten Speichers der begrenzende Faktor.

Wenn Sie ActiveMQ im persistenten Modus verwenden, wird normalerweise in den Speicher geschrieben, wenn entweder weniger Konsumenten vorhanden sind oder wenn die Konsumenten langsam sind. Im nicht-persistenten Modus wird bei langsamen Konsumenten auch in den Speicher geschrieben, wenn der Heap-Speicher der Broker-Instance voll ist.

Zum Bestimmen des besten Broker-Instance-Typs für Ihre Anwendung empfehlen wir, verschiedene Broker-Instance-Typen zu testen. Weitere Informationen finden Sie unter Broker instance types und auch Messung des Durchsatzes für Amazon MQ mithilfe des JMS Benchmarks.

Anwendungsfälle für größere Broker-Instance-Typen

Es gibt drei häufige Anwendungsfälle, wenn größere Broker-Instance-Typen den Durchsatz verbessern:

- Nicht-persistenter Modus - Wenn Ihre Anwendung weniger empfindlich gegenüber dem Verlust von Nachrichten während eines Broker-Instance-Failovers (z. B. bei der Übertragung von Sportergebnissen) ist, können Sie oft den nicht-persistenten Modus von ActiveMQ verwenden. In diesem Modus schreibt ActiveMQ Nachrichten nur dann in einen persistenten Speicher, wenn der Heap-Speicher der Broker-Instance voll ist. Systeme, die den nicht-persistenten Modus verwenden, können von der höheren Speichermenge und dem immer schnelleren Netzwerk profitierenCPU, die auf größeren Broker-Instance-Typen verfügbar sind.
- Schnelle Konsumenten - Wenn aktive Konsumenten verfügbar sind und das `concurrentStoreAndDispatchQueues`-Flag aktiviert ist, erlaubt ActiveMQ den direkten Nachrichtenfluss vom Produzenten zum Konsumenten, ohne Nachrichten an den Speicher zu senden (sogar im persistenten Modus). Wenn Ihre Anwendung Nachrichten schnell abrufen kann (oder wenn Sie Ihre Konsumenten entsprechend entwerfen können), kann Ihre Anwendung von einem größeren Broker-Instance-Typ profitieren. Damit Ihre Anwendung Nachrichten schneller abrufen kann, fügen Sie zu Ihren Anwendungs-Instances Konsumenten-Threads hinzu oder skalieren Sie Ihre Anwendungs-Instances vertikal oder horizontal nach oben.
- Als Stapel verarbeitete Transaktionen - Wenn Sie den persistenten Modus verwenden und mehrere Nachrichten pro Transaktion senden, können Sie durch Verwendung größerer Broker-Instance-Typen einen insgesamt höheren Durchsatz erzielen. Weitere Informationen finden Sie unter Sollte ich Transaktionen verwenden? in der Apache ActiveMQ-Dokumentation.

Auswählen des richtigen Broker-Speichertyps für den besten Durchsatz

Verwenden Sie Amazon, um die Vorteile der hohen Haltbarkeit und Replikation über mehrere Availability Zones hinweg zu nutzenEFS. Verwenden Sie Amazon, um von niedriger Latenz und hohem Durchsatz zu profitierenEBS. Weitere Informationen finden Sie unter Storage.

Korrekte Konfiguration Ihres Netzwerk von Brokern

Wenn Sie ein Netzwerk von Brokern erstellen, konfigurieren Sie es korrekt für Ihre Anwendung:

- Persistenten Modus aktivieren - Da (im Vergleich zu seinen Mitbewerbern) jede Broker-Instance wie ein Produzent oder ein Verbraucher agiert, bieten Netzwerke von Brokern keine verteilte Replikation von Nachrichten. Der erste Broker, der als Verbraucher auftritt, erhält eine Nachricht und verbleibt im Speicher. Dieser Broker sendet eine Bestätigung an den Produzenten und leitet die Nachricht an den nächsten Broker weiter. Wenn der zweite Broker die Persistenz der Nachricht bestätigt, löscht der erste Broker die Nachricht.

  Wenn der persistente Modus deaktiviert ist, bestätigt der erste Broker den Produzenten, ohne die Nachricht persistent im Speicher abzulegen. Weitere Informationen finden Sie unter Replicated Message Store und What is the difference between persistent and non-persistent delivery? in der Apache ActiveMQ-Dokumentation.
- Deaktivieren Sie Advisory Messages für Broker-Instances nicht - Weitere Informationen finden Sie unter Advisory Message in der Apache ActiveMQ-Dokumentation.
- Keine Multicast-Broker-Erkennung verwenden - Amazon MQ unterstützt die Brokererkennung über Multicast nicht. Weitere Informationen finden Sie unter What is the difference between discovery, multicast, and zeroconf? in der Apache ActiveMQ-Dokumentation.

Vermeiden von langsamen Neustarts durch Wiederherstellung vorbereiteter XA-Transaktionen

ActiveMQ unterstützt verteilte (XA-)Transaktionen. Zu wissen, wie ActiveMQ XA-Transaktionen verarbeitet, kann hilfreich sein, um langsame Wiederherstellungszeiten bei Broker-Neustarts und Failovers in Amazon MQ zu vermeiden.

Nicht aufgelöste vorbereitete XA-Transaktionen werden bei jedem Neustart erneut wiedergegeben. Wenn diese weiterhin nicht aufgelöst werden, wächst ihre Anzahl mit der Zeit weiter an, was die zum Starten des Brokers benötigte Zeit erheblich erhöht. Dies wirkt sich auf die Neustart- und Failover-Zeit

aus. Sie müssen diese Transaktionen mit einem `commit()` oder einem `rollback()` auflösen, damit sich die Leistung im Laufe der Zeit nicht verschlechtert.

Um Ihre ungelösten vorbereiteten XA-Transaktionen zu überwachen, können Sie die `JournalFilesForFastRecovery` Metrik in Amazon CloudWatch Logs verwenden. Wenn diese Zahl ansteigt oder ständig höher als 1 ist, sollten Sie Ihre nicht aufgelösten Transaktionen mit einem Code wie in dem folgenden Beispiel wiederherstellen. Weitere Informationen finden Sie unter Kontingente in Amazon MQ.

Der folgende Beispiel-Code führt Sie durch vorbereitete XA-Transaktionen und schließt sie mit einem `rollback()` ab.

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

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

public class RecoverXaTransactions {
    private static final ActiveMQXAConnectionFactory ACTIVE_MQ_CONNECTION_FACTORY;
    final static String WIRE_LEVEL_ENDPOINT =
            "tcp://localhost:61616";;
    static {
        final String activeMqUsername = "MyUsername123";
        final String activeMqPassword = "MyPassword456";
        ACTIVE_MQ_CONNECTION_FACTORY = new
 ActiveMQXAConnectionFactory(activeMqUsername, activeMqPassword, WIRE_LEVEL_ENDPOINT);
        ACTIVE_MQ_CONNECTION_FACTORY.setUserName(activeMqUsername);
        ACTIVE_MQ_CONNECTION_FACTORY.setPassword(activeMqPassword);
    }

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

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

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

In einem realen Szenario können Sie Ihre vorbereiteten XA-Transaktionen mithilfe Ihres XA Transaktionsmanagers überprüfen. Anschließend können Sie entscheiden, ob die Verarbeitung der einzelnen vorbereiteten Transaktionen mit einem `rollback()` oder einem `commit()` erfolgen soll.

Amazon MQ für RabbitMQ verwenden

Mit Amazon MQ ist es ganz einfach, einen Message Broker mit den Computing- und Speicherressourcen zu erstellen, die Ihren Anforderungen entsprechen. Sie können Broker mit dem, Amazon MQ REST API oder dem erstellen AWS Management Console, verwalten und löschen. AWS Command Line Interface

Dieser Abschnitt beschreibt die Grundelemente eines Message Brokers für ActiveMQ- und RabbitMQ-Engine-Typen, listet verfügbare Amazon MQ -Broker-Instance-Typen und deren Status auf und bietet einen Überblick über die Broker-Architektur und -Konfigurationsoptionen.

Weitere Informationen zu Amazon MQ REST APIs finden Sie in der Amazon MQ MQ-Referenz REST API.

Amazon MQ für RabbitMQ-Broker

Was ist ein Amazon MQ for RabbitMQ Broker?

Ein Broker ist eine Message-Broker-Umgebung, die auf Amazon MQ ausgeführt wird. Dies ist der Grundblock für Amazon MQ. Die kombinierte Beschreibung der Broker-Instanceclass(`m5`,`t3`) undsize(`large`,`micro`) ist einBroker-Instance-Typ(zum Beispiel`mq.m5.large`). Weitere Informationen finden Sie unter Broker instance types.

- Ein Single-Instance-Broker besteht aus einem Broker in einer Availability Zone hinter einem Network Load Balancer (NLB). Der Broker kommuniziert mit Ihrer Anwendung und mit einem EBS Amazon-Speichervolume.
- Ein Cluster-Bereitstellung ist eine logische Gruppierung von drei RabbitMQ-Broker-Knoten hinter einem Network Load Balancer, wobei jeder Benutzer, Warteschlangen und ein verteilter Status über mehrere Availability Zones (AZ) verfügt.

Weitere Informationen finden Sie unter Bereitstellungsoptionen für Amazon MQ für RabbitMQ-Broker.

Sie können automatische Upgrades auf Unterversionen aktivieren, damit Upgrades auf neue Unterversionen der Broker-Engine ausgeführt werden, sobald neue Versionen von RabbitMQ-Engine veröffentlicht werden. Automatische Upgrades erfolgen während des Wartungsfensters, das durch den Wochentag, die Tageszeit (im 24-Stunden-Format) und die Zeitzone (UTCstandardmäßig) definiert wird.

Unterstützte Protokolle

Sie können auf Ihre RabbitMQ-Broker zugreifen, indem Sie eine beliebige Programmiersprache verwenden, die RabbitMQ unterstützt, und indem Sie die folgenden Protokolle aktivieren: TLS

- AMQP(0-9-1)

Listener-Ports

Von Amazon MQ verwaltete RabbitMQ-Broker unterstützen die folgenden Listener-Ports für Konnektivität auf Anwendungsebene sowie Client-Verbindungen über `amqps` die RabbitMQ-Webkonsole und das Management. API

- Listener-Port — Wird für Verbindungen verwendet, die über Secure hergestellt werden. `5671` AMQP URL Bei einem Broker mit Broker-ID`b-c8352341-ec91-4a78-ad9c-a43f23d325bb`, der in der `us-west-2` Region eingesetzt wird, ist das Folgende die vollständige Adresse des Brokers amqpURL:`b-c8352341-ec91-4a78-ad9c-a43f23d325bb.mq.us-west-2.amazonaws.com:5671`.
- Listener-Ports `443` und `15671` — Beide Listener-Ports können austauschbar verwendet werden, um über die RabbitMQ-Webkonsole oder das Management auf einen Broker zuzugreifen. API

Attribute

Ein RabbitMQ-Broker verfügt über mehrere Attribute:

- Ein Name. Beispiel, `MyBroker`.
- Eine ID. Beispiel, `b-1234a5b6-78cd-901e-2fgh-3i45j6k178l9`.
- Ein Amazon-Ressourcenname (ARN). Beispiel, `arn:aws:mq:us-east-2:123456789012:broker:MyBroker:b-1234a5b6-78cd-901e-2fgh-3i45j6k178l9`.
- Eine RabbitMQ-Webkonsole. URL Beispiel, `https://b-1234a5b6-78cd-901e-2fgh-3i45j6k178l9-1.mq.us-east-2.amazonaws.com`.

  Weitere Informationen finden Sie unter RabbitMQ Webkonsole in der RabbitMQ-Dokumentation.
- Ein sicherer Endpunkt. AMQP Beispiel, `amqps://b-1234a5b6-78cd-901e-2fgh-3i45j6k178l9-1.mq.us-east-2.amazonaws.com`.

Eine vollständige Liste der Broker-Attribute finden Sie im Folgenden in der Amazon MQ REST API MQ-Referenz:

- RESTVorgangs-ID: Makler
- RESTVorgangs-ID: Makler
- RESTVorgangs-ID: Neustart des Brokers

Amazon MQ für RabbitMQ-Broker-Benutzer

Jeder AMQP 0-9-1-Client-Verbindung ist ein Benutzer zugeordnet, der authentifiziert werden muss. Jede Clientverbindung zielt auch auf einen virtuellen Host (vhost) ab, für den der Benutzer über eine Reihe von Berechtigungen verfügen muss. Ein Benutzer kann die Berechtigung haben, Warteschlangen und Exchanges in einem Vhost zu konfigurieren,schreiben, und zu lesen. Benutzeranmeldeinformationen und der Ziel-vhost werden bei der Verbindungsherstellung angegeben.

Wenn Sie zum ersten Mal einen Broker für Amazon MQ für RabbitMQ erstellen, verwendet Amazon MQ die von Ihnen angegebenen Anmeldeinformationen, um einen RabbitMQ-Benutzer mit dem `administrator`-Tag zu erstellen. Sie können dann Benutzer über die RabbitMQ-Verwaltung oder die RabbitMQ-Webkonsole hinzufügen und verwalten. API Sie können auch die RabbitMQ-Webkonsole oder die Verwaltung API verwenden, um Benutzerberechtigungen und Tags festzulegen oder zu ändern.

> **Note**
>
> RabbitMQ-Benutzer werden nicht über die Amazon MQ MQ-Benutzer gespeichert oder angezeigt. API

> **Important**
>
> Amazon MQ for RabbitMQ unterstützt den Benutzernamen „guest“ nicht und löscht das Standard-Gastkonto, wenn Sie einen neuen Broker erstellen. Amazon MQ löscht außerdem regelmäßig alle vom Kunden erstellten Konten mit dem Namen „Gast“.

Verwenden Sie den folgenden API Endpunkt und den folgenden AnfragetextAPI, um einen neuen Benutzer mit der RabbitMQ-Verwaltung zu erstellen. Ersetzen *username* and *password* mit Ihren neuen Anmeldedaten.

```
PUT /api/users/username HTTP/1.1

            {"password":"password","tags":"administrator"}
```

Important

- Fügen Sie den Benutzernamen von Brokern keine personenbezogenen Daten (PII) oder andere vertrauliche oder sensible Informationen hinzu. Broker-Benutzernamen sind für andere AWS Dienste zugänglich, einschließlich CloudWatch Logs. Broker-Benutzernamen sind nicht für private oder sensible Daten gedacht.
- Wenn Sie das Administratorpasswort vergessen haben, das Sie bei der Erstellung des Brokers festgelegt haben, können Sie Ihre Anmeldeinformationen nicht zurücksetzen. Wenn Sie mehrere Administratoren erstellt haben, können Sie sich über einen anderen Administratorbenutzer anmelden und Ihre Anmeldeinformationen zurücksetzen oder neu erstellen. Wenn Sie nur einen Administratorbenutzer haben, müssen Sie den Broker löschen und einen neuen mit neuen Anmeldeinformationen erstellen. Wir empfehlen, Nachrichten zu lesen oder zu sichern, bevor Sie den Broker löschen.

Der `tags`-Schlüssel ist obligatorisch und besteht aus einer durch Kommas getrennten Liste von Tags für den Benutzer. Amazon MQ unterstützt `administrator`-,`management`-, `monitoring`- und `policymaker`-Benutzer-Tags.

Sie können Berechtigungen für einen einzelnen Benutzer festlegen, indem Sie den folgenden API Endpunkt und den folgenden Anforderungstext verwenden. Ersetzen *vhost* and *username* mit Ihren Informationen. Für den Standard-vhost`/`, verwenden Sie`%2F`.

```
PUT /api/permissions/vhost/username HTTP/1.1

            {"configure":".*","write":".*","read":".*"}
```

> **Note**
>
> Die Schlüssel `configure`, `read` und `write` sind alle Pflichtfelder.

Die Verwendung des Platzhalters `.*`-Wert gewährt dieser Vorgang dem Benutzer Lese-, Schreib- und Konfigurationsberechtigungen für alle Warteschlangen im angegebenen vhost. Weitere Informationen zur Verwaltung von Benutzern über die RabbitMQ-Verwaltung finden Sie unter RabbitMQ-VerwaltungAPI. HTTP API

Standardwerte für Amazon MQ für RabbitMQ Broker

Wenn Sie einen Amazon MQ für RabbitMQ Broker erstellen, wendet Amazon MQ einen Standardsatz von Broker-Richtlinien und vhost-Limits an, um die Leistung Ihres Brokers zu optimieren. Amazon MQ wendet Vhost-Beschränkungen nur auf den Standardwert (`/`) vhost an. Amazon MQ wendet keine Standardrichtlinien auf neu erstellte vhosts an. Wir empfehlen, diese Standardwerte für alle neuen und bestehenden Broker beizubehalten. Sie können diese Standardwerte jedoch jederzeit ändern, überschreiben oder löschen.

Amazon MQ erstellt Richtlinien und Limits basierend auf dem Instance-Typ und dem Broker-Bereitstellungsmodus, den Sie beim Erstellen Ihres Brokers auswählen. Die Standardrichtlinien werden gemäß dem Bereitstellungsmodus wie folgt benannt:

- Einzelne Instance – `AWS-DEFAULT-POLICY-SINGLE-INSTANCE`
- Cluster-Bereitstellung–`AWS-DEFAULT-POLICY-CLUSTER-MULTI-AZ`

FürSingle-Instance-Brokerfestgelegt ist, legt Amazon MQ den Richtlinienprioritätswert auf`0`. Um den Standardprioritätswert zu überschreiben, können Sie eigene benutzerdefinierte Richtlinien mit höheren Prioritätswerten erstellen. Für Cluster-Bereitstellungen, setzt Amazon MQ den Prioritätswert auf1für Broker-Standardwerte fest. Um eine eigene benutzerdefinierte Richtlinie für Cluster zu erstellen, weisen Sie einen Prioritätswert zu, der größer als1 ist.

> **Note**
>
> In Clusterbereitstellungen `ha-mode` und `ha-sync-mode` Broker-Richtlinien sind für die klassische Spiegelung und Hochverfügbarkeit (HA) erforderlich.
> Wenn Sie die Standardeinstellung`AWS-DEFAULT-POLICY-CLUSTER-MULTI-AZ`-Richtlinie verwenden, verwendet Amazon MQ die`ha-all-AWS-OWNED-DO-NOT-DELETE`-Richtlinie

mit dem Prioritätswert`0`. Dadurch wird sichergestellt, dass die erforderlichen `ha-mode` und `ha-sync-mode`-Richtlinien weiterhin in Kraft sind. Wenn Sie Ihre eigene benutzerdefinierte Richtlinie erstellen, hängt Amazon MQ automatisch `ha-mode` und `ha-sync-mode` zu Ihren Richtliniendefinitionen an.

Themen

- Richtlinien- und Grenzbeschreibungen
- Empfohlene Standardwerte

Richtlinien- und Grenzbeschreibungen

In der folgenden Liste werden die Standardrichtlinien und -beschränkungen beschrieben, die Amazon MQ für einen neu erstellten Broker anwendet. Die Werte für`max-length`,`max-queues`, und `max-connections` variieren je nach Instance-Typ und Bereitstellungsmodus Ihres Brokers. Diese Werte werden im Feld Abschnitts Empfohlene Standardwerte erstellt.

- **`queue-mode: lazy`**(Richtlinie) — Aktiviert Lazy-Warteschlangen. Standardmäßig halten Warteschlangen einen In-Memory-Cache von Nachrichten, so dass der Broker Nachrichten so schnell wie möglich an Verbraucher senden kann. Dies kann dazu führen, dass der Broker der Speicher ausläuft und einen Alarm mit hohem Speicher auslöst. Lazy Queues versuchen, Nachrichten so früh wie möglich auf den Datenträger zu verschieben. Dies bedeutet, dass unter normalen Betriebsbedingungen weniger Meldungen im Speicher gespeichert werden. Amazon MQ für RabbitMQ kann mithilfe von Lazy Queues viel größere Messaging-Lasten und längere Warteschlangen unterstützen. Beachten Sie, dass in bestimmten Anwendungsfällen Broker mit faulen Warteschlangen möglicherweise geringfügig langsamer ausgeführt werden. Dies liegt daran, dass Nachrichten vom Datenträger zu Broker verschoben werden, anstatt Nachrichten aus einem In-Memory-Cache zu übermitteln.

  > **Bereitstellungsmodi**
  >
  > Ein Single-Instance-Cluster

- **`max-length: number-of-messages`**(Richtlinie) — Legt ein Limit für die Anzahl der Nachrichten in einer Warteschlange fest. In Clusterbereitstellungen verhindert das Limit die angehaltene Warteschlangensynchronisierung in Fällen wie Broker-Neustarts oder im Anschluss an ein Wartungsfenster.

> **Bereitstellungsmodi**
>
> Cluster

- **`overflow: reject-publish`**(policy) — Erzwingt Warteschlangen mit einem`max-length`Um neue Nachrichten abzulehnen, nachdem die Anzahl der Nachrichten in der Warteschlange den`max-length`Wert erreicht. Um sicherzustellen, dass Nachrichten nicht verloren gehen, wenn sich eine Warteschlange in einem Überlaufzustand befindet, müssen Clientanwendungen, die Nachrichten an den Broker Herausgeber bestätigt implementieren. Weitere Informationen zur Implementierung von Publisher-Bestätigungen finden Sie unter Herausgeber bestätigt auf der RabbitMQ-Website.

> **Bereitstellungsmodi**
>
> Cluster

- **`max-queues: number-of-queues-per-vhost`**(vhost-Limit) — Legt das Limit für die Anzahl der Warteschlangen in einem Broker fest. Ähnlich wie bei `max-length`-Richtliniendefinition verhindert die Begrenzung der Anzahl der Warteschlangen in Clusterbereitstellungen die angehaltene Warteschlangensynchronisierung nach Broker-Neustarts oder Wartungsfenstern. Durch die Begrenzung von Warteschlangen wird auch eine übermäßige CPU Nutzung der Warteschlangen verhindert.

> **Bereitstellungsmodi**
>
> Ein Single-Instance-Cluster

- **`max-connections: number-of-connections-per-vhost`**(vhost-Limit) — Legt das Limit für die Anzahl der Clientverbindungen zum Broker fest. Die Begrenzung der Anzahl an Verbindungen gemäß den empfohlenen Werten verhindert eine übermäßige Broker-Speicherauslastung, die dazu führen könnte, dass der Broker einen Speicher-Alarm auslöst und Operationen pausiert.

> **Bereitstellungsmodi**
>
> Ein Single-Instance-Cluster

Empfohlene Standardwerte

> **Note**
>
> Die `max-length` und `max-queue` Standardlimits werden basierend auf einer durchschnittlichen Nachrichtengröße von 5 kB getestet und ausgewertet. Wenn Ihre Nachrichten deutlich größer als 5 kB sind, müssen Sie die`max-length` und `max-queue`-Beschränkungen.

In der folgenden Tabelle finden Sie die Standardgrenzwerte für einen neu erstellten Broker. Amazon MQ wendet diese Werte entsprechend dem Instance-Typ und dem Bereitstellungsmodus des Brokers an.

Instance-Typ	Bereitstellungsmodus	**max-length**	**max-queues**	**max-connections**
t3.micro	Single-Instance	N/A	500	500
m5.large	Single-Instance	N/A	20 000	4.000
	Cluster	8.000.000	4.000	15 000
m5.xlarge	Single-Instance	N/A	30 000	8 000
	Cluster	9.000.000	5,000	20 000
m5.2xlarge	Single-Instance	N/A	60 000	15 000
	Cluster	10 000 000	6 000	40 000
m5.4xlarge	Single-Instance	N/A	150.000	30 000
	Cluster	12.000.000	10.000	100 000

Broker-Instance-Typen von Amazon MQ für RabbitMQ

> **Important**
>
> Sie können einen Broker nicht von einem `mq.m5.`-Instance-Typ zu einem `mq.t3.micro`-Instance-Typ herunterstufen.

Instance-Typ	v CPU	Arbeitsspeicher (GiB)	Netzwerkleistung	Anwendungsfall
`mq.t3.micro`	2	1	Niedrig	Bewertung **Important** Der `mq.t3.micro` -Instance -Typ unterstützt die Cluster-Bereitstellung nicht.
`mq.m5.large`	2	8	Hoch	Produktion
`mq.m5.xlarge`	4	16	Hoch	Produktion
`mq.m5.2xlarge`	8	32	Hoch	
`mq.m5.4xlarge`	16	64	Hoch	

Größenrichtlinien für Amazon MQ für RabbitMQ

Sie können den Broker-Instance-Typ wählen, der Ihre Anwendung am besten unterstützt. Bei der Auswahl eines Instance-Typs ist es wichtig, Faktoren zu berücksichtigen, die sich auf die Leistung des Brokers auswirken:

- die Anzahl der Clients und Warteschlangen
- die Menge der gesendeten Nachrichten
- Nachrichten, die im Speicher aufbewahrt werden
- redundante Nachrichten

Kleinere Broker-Instance-Typen (`t3.micro`) werden nur zum Testen der Anwendungsleistung empfohlen. Wir empfehlen größere Broker-Instance-Typen (`m5.large`und höher) für die Produktion von Clients und Warteschlangen, hohen Durchsatz, Nachrichten im Speicher und redundante Nachrichten.

Es ist wichtig, Ihre Broker zu testen, um den geeigneten Instance-Typ und die Größe für Ihre Workload-Messaging-Anforderungen zu ermitteln. Verwenden Sie die folgenden Größenrichtlinien, um den für Ihre Anwendung am besten geeigneten Instance-Typ zu ermitteln.

Richtlinien zur Größenbestimmung für die Bereitstellung einer einzelnen Instanz

Die folgende Tabelle zeigt die maximalen Grenzwerte für jeden Instance-Typ für Single-Instance-Broker.

Instance-Typ	Verbindungen	Kanäle	Warteschlangen	Verbraucher pro Kanal	Schaufeln
t3.micro	500	1.500	2.500	1.000	150
m5.large	5,000	15 000	30 000	1.000	250
m5.xlarge	10.000	30 000	60 000	1.000	500
m5.2xlarge	20 000	60 000	120.000	1.000	1.000
m5.4xlarge	40 000	120.000	240.000	1.000	2.000

Richtlinien zur Größenbestimmung für die Clusterbereitstellung

Die folgende Tabelle zeigt die maximalen Grenzwerte für jeden Instance-Typ für Cluster-Broker.

Instance-Typ	Verbindungen	Kanäle	Warteschlangen	Verbraucher pro Kanal	Schaufeln
m5.large	15 000	45.000	10.000	1.000	150
m5.xlarge	30 000	90.000	15 000	1.000	300
m5.2xlarge	60 000	180 000	20 000	1.000	600
m5.4xlarge	120.000	360 000	30 000	1.000	1200

Die Verbindungs-, Kanal- und Schaufelgrenzwerte werden pro Knoten angewendet. Die genauen Grenzwerte für einen Cluster-Broker können niedriger als der angegebene Wert sein, abhängig von der Anzahl der verfügbaren Knoten und davon, wie RabbitMQ die Ressourcen auf die verfügbaren Knoten verteilt.

Fehlermeldungen

Die folgenden Fehlermeldungen werden zurückgegeben, wenn Grenzwerte überschritten werden. Alle Werte basieren auf den Grenzwerten für `m5.large` einzelne Instanzen.

> **Note**
>
> Die Fehlercodes für die folgenden Meldungen können sich je nach verwendeter Client-Bibliothek ändern.

Connection (Verbindung)

```
ConnectionClosedByBroker 500 "NOT_ALLOWED - connection refused: node
connection limit (500) is reached"
```

Channel

```
ConnectionClosedByBroker 1500 "NOT_ALLOWED - number of channels opened on
node 'rabbit@ip-10-0-23-173.us-west-2.compute.internal' has reached the
maximum allowed limit of (15,000)"
```

Verbraucher

```
ConnectionClosedByBroker: (530, 'NOT_ALLOWED - reached maximum (1,000) of
consumers per channel')
```

> **Note**
>
> Die folgenden Fehlermeldungen verwenden das HTTP API Management-Format.

Warteschlange

```
{"error":"bad_request","reason":"cannot declare queue 'my_queue': queue
limit in cluster (30,000) is reached"}]
```

Schaufel

```
{"error":"bad_request","reason":"Validation failed\n\ncomponent shovel is
limited to 250 per node\n"}
```

Gespenst

```
{"error":"bad_request","reason":"cannot create vhost 'my_vhost': vhost
limit of 4,000 is reached"}
```

Plugins für Amazon MQ für RabbitMQ

Amazon MQ for RabbitMQ unterstützt das RabbitMQ-Verwaltungs-Plugin, das die Verwaltung und die RabbitMQ-Webkonsole unterstützt. API Sie können die Webkonsole und das Management verwenden, um Broker-Benutzer und -Richtlinien zu erstellen und zu verwalten. API

Neben dem Management Plugin unterstützt Amazon MQ für RabbitMQ auch die folgenden Plug-ins.

Themen

- Shovel Plugin
- Federation Plugin
- Consistent Hash Exchange Plugin

Shovel Plugin

Von Amazon MQ verwaltete Makler unterstützen die RabbitMQ Shovel, sodass Sie Nachrichten aus Warteschlangen und Exchanges auf einer Broker-Instance in eine andere verschieben können. Sie können Shovel verwenden, um lose gekoppelte Broker zu verbinden und Nachrichten von Knoten mit schwereren Nachrichtenladungen zu verteilen.

Von Amazon MQ verwaltete RabbitMQ Broker unterstützen dynamische Shovels. Dynamische Shovels werden mit Laufzeitparametern konfiguriert und können jederzeit programmgesteuert über eine Clientverbindung gestartet und gestoppt werden. Mithilfe des RabbitMQ-Managements können Sie beispielsweise eine PUT Anfrage an den folgenden API Endpunkt erstellenAPI, um eine dynamische Schaufel zu konfigurieren. Im Beispiel kann `{vhost}` durch den Namen des vhost des Brokers und `{name}` durch den Namen der neuen dynamischen Shovel ersetzt werden.

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

Im Anforderungstext müssen Sie entweder eine Warteschlange oder einen Exchange angeben, aber nicht beides. In diesem Beispiel unten wird eine dynamische Shovel zwischen einer in `src-queue` angegebenen lokalen Warteschlange und einer in `dest-queue` definierten Remote-Warteschlange konfiguriert. Auf ähnliche Weise können Sie die Parameter `src-exchange` und `dest-exchange` verwenden, um eine Shovel zwischen zwei Exchanges zu konfigurieren.

```
{
                "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**
>
> Sie können die Shovel zwischen Warteschlangen oder Exchanges nicht konfigurieren, wenn das Shovel-Ziel ein privater Broker ist. Sie können die Shovel nur zwischen Warteschlangen

oder Exchanges in öffentlichen Brokern oder zwischen einer Quelle in einem privaten Broker und einem Ziel in einem öffentlichen Broker konfigurieren.

Weitere Informationen zur Verwendung dynamischer Shovels finden Sie unter dem RabbitMQ Dynamic Shovel Plugin.

> **Note**
>
> Amazon MQ unterstützt die Verwendung statischer Shoveln nicht.

Federation Plugin

Amazon MQ unterstützt Verbund-Exchange und -Warteschlangen. Mit Verbund können Sie den Nachrichtenfluss zwischen Warteschlangen, Exchanges und Verbrauchern auf separaten Brokern replizieren. Verbundwarteschlangen und Exchanges verwenden point-to-point Links, um Verbindungen zu Kollegen in anderen Brokern herzustellen. Während Verbund-Exchanges Nachrichten standardmäßig einmal weiterleiten, können Verbundwarteschlangen Nachrichten beliebig oft verschieben, wie es von den Verbrauchern benötigt wird.

Sie können einen Verbund verwenden, um einen Downstream--Broker zu ermöglichen, eine Nachricht von einem Exchange oder einer Warteschlange auf einen Upstream-Broker zu verwenden. Sie können den Verbund auf nachgeschalteten Brokern aktivieren, indem Sie die RabbitMQ-Webkonsole oder das Management verwenden. API

> **Important**
>
> Sie können den Verbund nicht konfigurieren, wenn sich die Upstream-Warteschlange oder der Exchange in einem privaten Broker befindet. Sie können nur den Verbund zwischen Warteschlangen oder Exchanges in öffentlichen Brokern oder zwischen einer Upstream-Warteschlange oder einem Exchange in einem öffentlichen Broker und einer Downstream-Warteschlange oder einer Börse in einem privaten Broker konfigurieren.

Mithilfe der Verwaltung können Sie den API Verbund beispielsweise wie folgt konfigurieren.

- Konfigurieren Sie einen oder mehrere Upstreams, die Verbundverbindungen zu anderen Knoten definieren. Sie können Verbundverbindungen mithilfe der RabbitMQ-Webkonsole oder der

Verwaltung definieren. API Mithilfe der Verwaltung API können Sie eine `POST` Anfrage an `/api/parameters/federation-upstream/%2f/`*`my-upstream`* mit dem folgenden Anfragetext erstellen.

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

- Konfigurieren Sie eine Richtlinie, damit Ihre Warteschlangen oder Exchanges miteinander verbunden werden können. Sie können Richtlinien mithilfe der RabbitMQ-Webkonsole oder der Verwaltung konfigurieren. API Mithilfe der Verwaltung API können Sie eine `POST` Anfrage an `/api/policies/%2f/federate-me` mit dem folgenden Anfragetext erstellen.

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

> **Note**
>
> Der Anforderungstext nimmt an, dass Exchanges auf dem Server mit `amq` beginnen. Verwenden von regulären Ausdrücken `^amq\.` stellt sicher, dass der Verbund für alle Börsen aktiviert ist, deren Namen mit „amq“ beginnen. Die Exchanges auf Ihrem RabbitMQ-Server können unterschiedlich benannt werden.

Weitere Informationen zum Konfigurieren des Federation Plugins finden Sie unter RabbitMQ Federation Plugin.

Consistent Hash Exchange Plugin

Standardmäßig unterstützt Amazon MQ für RabbitMQ das Exchange Plug-in „Consistent Hash“. Consistent Hash tauscht Routing-Nachrichten an Warteschlangen aus, basierend auf einem Hash-Wert, der aus dem Routing-Schlüssel einer Nachricht berechnet wird. Angesichts eines ziemlich gleichmäßigen Routingschlüssels können Cosistent Hash Exchanges Nachrichten zwischen Warteschlangen relativ gleichmäßig verteilen.

Bei Warteschlangen, die an einen konsistenten Hash-Austausch gebunden sind, ist der Bindungsschlüssel a number-as-a-string , der das Bindungsgewicht jeder Warteschlange bestimmt. Warteschlangen mit einer höheren Bindungsstärke erhalten eine proportional höhere Verteilung von Nachrichten aus dem Cosistent Hash Exchange, an den sie gebunden sind. In einer Consistent Hash Exchange-Topologie können Publisher einfach Nachrichten in der Exchange veröffentlichen, aber

Verbraucher müssen explizit konfiguriert werden, um Nachrichten aus bestimmten Warteschlangen zu verwenden.

Weitere Informationen zu Consistent Hash Exchanges finden Sie auf der Website unter RabbitMQ Consistent Hash Exchange Type. GitHub

Anwenden von Richtlinien auf Amazon MQ für RabbitMQ

Sie können benutzerdefinierte Richtlinien und Beschränkungen mit den von Amazon MQ empfohlenen Standardwerten anwenden. Wenn Sie die empfohlenen Standardrichtlinien und -grenzwerte gelöscht haben und sie neu erstellen möchten, oder Sie zusätzliche Vhosts erstellt haben und die Standardrichtlinien und -grenzwerte auf Ihre neuen Vhosts anwenden möchten, können Sie die folgenden Schritte ausführen.

Important

Um die folgenden Schritte ausführen zu können, benötigen Sie einen Amazon MQ -Broker-Benutzer mit Administratorberechtigungen. Sie können den Administratorbenutzer verwenden, der beim ersten Erstellen des Brokers erstellt wurde, oder einen anderen Benutzer, den Sie später erstellt haben. Die folgende Tabelle enthält die erforderlichen Administratorbenutzer-Tag und Berechtigungen als reguläre Ausdrücke (regexp) Muster.

Tags	Lesen Sie regexp	Konfigurieren von regexp	REGEXP-Schreiben
`administrator`	`.*`	`.*`	`.*`

Weitere Informationen zum Erstellen von RabbitMQ-Benutzern und zum Verwalten von Benutzer-Tags und -berechtigungen finden Sie unterAmazon MQ für RabbitMQ-Broker-Benutzer.

So wenden Sie Standardrichtlinien und virtuelle Host-Limits mit der RabbitMQ-Webkonsole an

1. Melden Sie sich bei der Amazon MQ-Konsole an.
2. Wählen Sie im linken Navigationsbereich die Option Broker aus.

3. Wählen Sie in der Broker-Liste den Namen des Brokers aus, auf den Sie die neue Richtlinie anwenden möchten.
4. Wählen Sie auf der Seite mit den Broker-Details im Abschnitt Verbindungen die RabbitMQ-Webkonsole aus. URL Die RabbitMQ-Webkonsole wird in einer neuen Browserregisterkarte oder -fenster geöffnet.
5. Melden Sie sich mit Ihrem Broker-Administratornamen und -Passwort an der RabbitMQ-Webkonsole an.
6. Wählen Sie in der RabbitMQ-Webkonsole oben auf der Seite die OptionAdmin.
7. Klicken Sie auf derAdminWählen Sie im rechten Navigationsbereich die OptionRichtlinien.
8. Klicken Sie auf derRichtlinienkönnen Sie eine Liste der aktuellen Broker-Benutzerrichtlinien sehen. UnterBenutzerrichtlinienerweitern SieRichtlinie hinzufügen/aktualisieren.
9. Um eine neue Broker-Richtlinie zu erstellen, tun Sie das Folgende unterRichtlinie hinzufügen/aktualisieren:

    a. FürVirtueller Host, wählen Sie in der Dropdown-Liste den Namen des Vhosts aus, dem die Richtlinien angehängt werden sollen. Um den Standard-Vhost auszuwählen, wählen Sie/.

    > **Note**
    >
    > Wenn Sie keine zusätzlichen Vhosts erstellt haben, wird dieVirtueller Hostwird in der RabbitMQ-Konsole nicht angezeigt, und die Richtlinien werden nur auf den Standard-vhost angewendet.

    b. Geben Sie unter Name einen Namen für Ihre Richtlinie ein, z. B. **`policy-defaults`**.

    c. Für Pattern geben Sie das regexp-Muster ein**`.*`**, damit die Richtlinie mit allen Warteschlangen auf dem Broker übereinstimmt.

    d. FürÜbernehmen von, wählen Sie Tauschen von Warteschlangen aus der Dropdown-Liste.

    e. Für Priority (Priorität), geben Sie eine Ganzzahl ein, die größer ist als alle anderen Richtlinien, die auf den vhost angewendet werden. Sie können jederzeit genau einen Satz von Richtliniendefinitionen auf RabbitMQ-Warteschlangen und -Austauschvorgänge anwenden. RabbitMQ wählt die Matching-Policy mit dem höchsten Prioritätswert . Weitere Informationen zu Richtlinienprioritäten und zum Kombinieren von Richtlinien finden Sie unter Richtlinien in der Dokumentation zu RabbitMQ Server.

    f. Für Definition, fügen Sie die folgenden Schlüssel-Wert-Paare hinzu:

    - **`queue-mode=lazy`**. Klicken Sie aufZeichenfolgeaus der Dropdown-Liste.

- **`overflow=reject-publish`**. Klicken Sie aufZeichenfolgeaus der Dropdown-Liste.

  > **Note**
  >
  > Gilt nicht für Single-Instance-Broker.

- **`max-length=`**. ***`number-of-messages`*** Ersetze *`number-of-messages`* mit dem von Amazon MQ empfohlenen Wert entsprechend der Instance-Größe und dem Bereitstellungsmodus des Brokers, z. B. **`8000000`** für einen `mq.m5.large` Cluster. Wählen Sie Number aus der Dropdown-Liste.

  > **Note**
  >
  > Gilt nicht für Single-Instance-Broker.

g. Wählen Sie Add / update policy.

10. Vergewissern Sie sich, dass die neue Richtlinie in der Liste derBenutzerrichtlinien.

> **Note**
>
> Für Cluster-Broker wendet Amazon MQ automatisch die`ha-mode: all`und`ha-sync-mode: automatic`-Definitionen.

11. Wählen Sie im Navigationsbereich die Option Limits aus.
12. Klicken Sie auf derEinschränkungenSie können eine Liste der aktuellenGrenzits für virtuelle Host. UnterGrenzits für virtuelle Hosterweitern SieFestlegen oder Aktualisieren eines virtuellen Hosts.
13. Um ein neues vhost-Limit zu erstellen, gene Sie unterFestlegen oder Aktualisieren eines virtuellen Hosts wie folgt vor:

    a. FürVirtueller Host, wählen Sie in der Dropdown-Liste den Namen des Vhosts aus, dem die Richtlinien angehängt werden sollen. Um den Standard-Vhost auszuwählen, wählen Sie/.

    b. FürLimit, wählen Siemax-connectionsaus den Dropdown-Optionen.

    c. FürValue, geben Sie den Amazon MQ Empfohlenenen Wertentsprechend der Instance-Größe und dem Bereitstellungsmodus des Brokers ein, z. B.**`15000`**für einen `mq.m5.large`-Cluster.

    d. Klicken Sie aufGrenzwert setzen/aktualisieren.

e. Wiederholen Sie die obigen Schritte und fürLimit, wählen Siemax-queuesaus den Dropdown-Optionen.

14. Vergewissern Sie sich, dass die neuen Grenzwerte in der Liste derGrenzits für virtuelle Host.

Um Standardrichtlinien und Grenzwerte für virtuelle Hosts mithilfe des RabbitMQ-Managements anzuwenden API

1. Melden Sie sich bei der Amazon MQ-Konsole an.
2. Wählen Sie im linken Navigationsbereich die Option Broker aus.
3. Wählen Sie in der Broker-Liste den Namen des Brokers aus, auf den Sie die neue Richtlinie anwenden möchten.
4. Beachten Sie auf der Broker-Seite im Abschnitt Verbindungen die RabbitMQ-Webkonsole. URL Dies ist der Broker-Endpunkt, den Sie in einer Anfrage verwenden. HTTP
5. Öffnen Sie ein neues Terminal- oder Befehlszeilenfenster Ihrer Wahl.
6. Um eine neue Broker-Richtlinie zu erstellen, geben Sie Folgendes ein`curl`-Befehl. Dieser Befehl nimmt an, dass eine Warteschlange auf der Standardeinstellung/vhost, der als`%2F` encodiert ist. Um die Richtlinie auf einen anderen Vhost anzuwenden, ersetzen Sie`%2F`durch den Vhost-Namen.

> **Note**
>
> Ersetzen *`username`* and *`password`* mit Ihren Administrator-Anmeldedaten. Ersetzen *`number-of-messages`* mit dem von Amazon MQ empfohlenen Wert entsprechend der Instance-Größe und dem Bereitstellungsmodus des Brokers. Ersetzen *`policy-name`* mit einem Namen für Ihre Richtlinie. Ersetzen *`broker-endpoint`* mit demURL, was Sie zuvor notiert haben.

```
curl -i -u username:password -H "content-type:application/json" -XPUT \
-d '{"pattern":".*", "priority":1, "definition":{"queue-mode":lazy,
 "overflow":"reject-publish", "max-length":"number-of-messages"}}' \
broker-endpoint/api/policies/%2F/policy-name
```

7. Um zu bestätigen, dass die neue Richtlinie den Benutzerrichtlinien Ihres Brokers hinzugefügt wird, geben Sie folgenden `curl`-Befehl ein, um alle Broker-Richtlinien aufzulisten.

```
curl -i -u username:password broker-endpoint/api/policies
```

8. Um ein neues `max-connections`virtuelles Host-Limit zu erstellen, geben Sie folgenden `curl`-Befehl ein. Dieser Befehl nimmt an, dass eine Warteschlange auf der Standardeinstellung/vhost, der als`%2F`. Um die Richtlinie auf einen anderen Vhost anzuwenden, ersetzen Sie`%2F`durch den Vhost-Namen.

> **Note**
>
> Ersetzen *`username`* and *`password`* mit Ihren Administrator-Anmeldedaten. Ersetzen *`max-connections`* mit dem von Amazon MQ empfohlenen Wert entsprechend der Instance-Größe und dem Bereitstellungsmodus des Brokers. Ersetzen Sie den Broker-Endpunkt durch denURL, den Sie zuvor notiert haben.

```
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. Um ein neus `max-queues` Virtual Host-Limit zu erstellen, wiederholen Sie den vorherigen Schritt, ändern Sie jedoch den curl-Befehl wie im Folgenden gezeigt.

```
curl -i -u username:password -H "content-type:application/json" -XPUT \
-d '{"value":"number-of-queues"}' \
broker-endpoint/api/vhost-limits/%2F/max-queues
```

10. Um zu bestätigen, dass die neuen Limits zu den virtuellen Host-Limits Ihres Brokers hinzugefügt werden, geben Sie Folgendes ein:`curl`, um alle virtuellen Host-Grenzwerte für Broker aufzulisten.

```
curl -i -u username:password broker-endpoint/api/vhost-limits
```

Bereitstellungsoptionen für Amazon MQ für RabbitMQ-Broker

RabbitMQ Broker können alsSingle-Instance-Brokeroder in einemCluster-Bereitstellung. Für beide Bereitstellungsmodi bietet Amazon MQ eine hohe Haltbarkeit, indem seine Daten redundant gespeichert werden.

Sie können auf Ihre RabbitMQ-Broker zugreifen, indem Sie eine beliebige Programmiersprache verwenden, die RabbitMQ unterstützt, und indem Sie die folgenden Protokolle aktivieren: TLS

- AMQP(0-9-1)

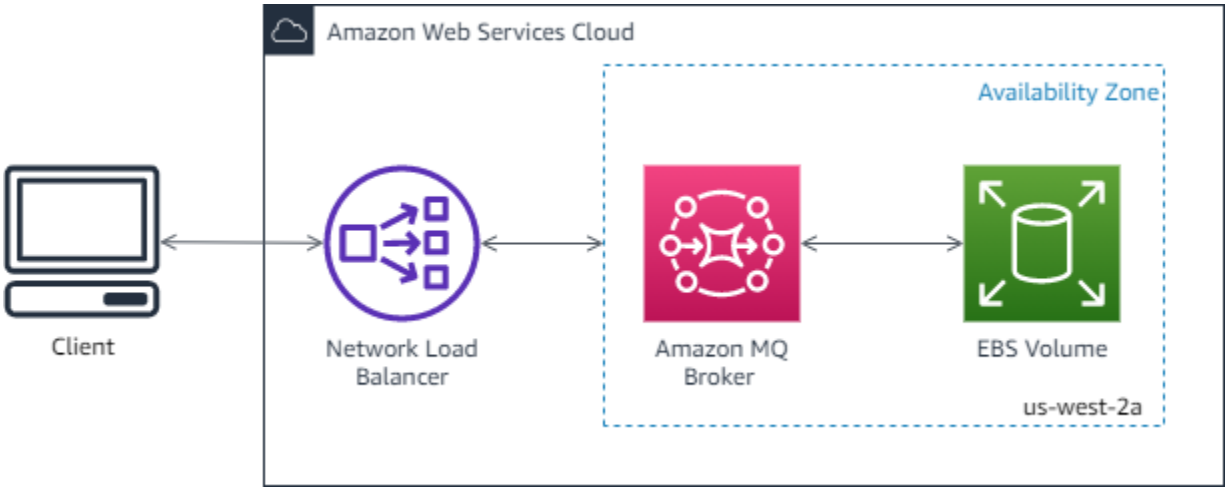
Themen

Option 1: Einzelinstanz-Broker Amazon MQ für RabbitMQ

Ein Single-Instance-Broker besteht aus einem Broker in einer Availability Zone hinter einem Network Load Balancer ()NLB. Der Broker kommuniziert mit Ihrer Anwendung und mit einem EBS Amazon-Speichervolumen. Amazon EBS bietet Speicher auf Blockebene, der für niedrige Latenz und hohen Durchsatz optimiert ist.

Durch die Verwendung eines Network Load Balancer wird sichergestellt, dass Ihr Amazon MQ for RabbitMQ Broker-Endpunkt unverändert bleibt, wenn die Broker-Instance während eines Wartungsfensters oder aufgrund von zugrunde liegenden Amazon-Hardwarefehlern ersetzt wird. EC2 Mit einem Network Load Balancer können Ihre Anwendungen und Benutzer weiterhin denselben Endpunkt verwenden, um eine Verbindung mit dem Broker herzustellen.

Das folgende Diagramm verdeutlicht einen Amazon MQ for RabbitMQ Single-Instance-Broker.



Option 2: Amazon MQ für die RabbitMQ-Clusterbereitstellung

Eine Cluster-Bereitstellung ist eine logische Gruppierung von drei RabbitMQ-Broker-Knoten hinter einem Network Load Balancer, wobei jeder Benutzer, Warteschlangen und ein verteilter Status über mehrere Availability Zones (AZ) verfügt.

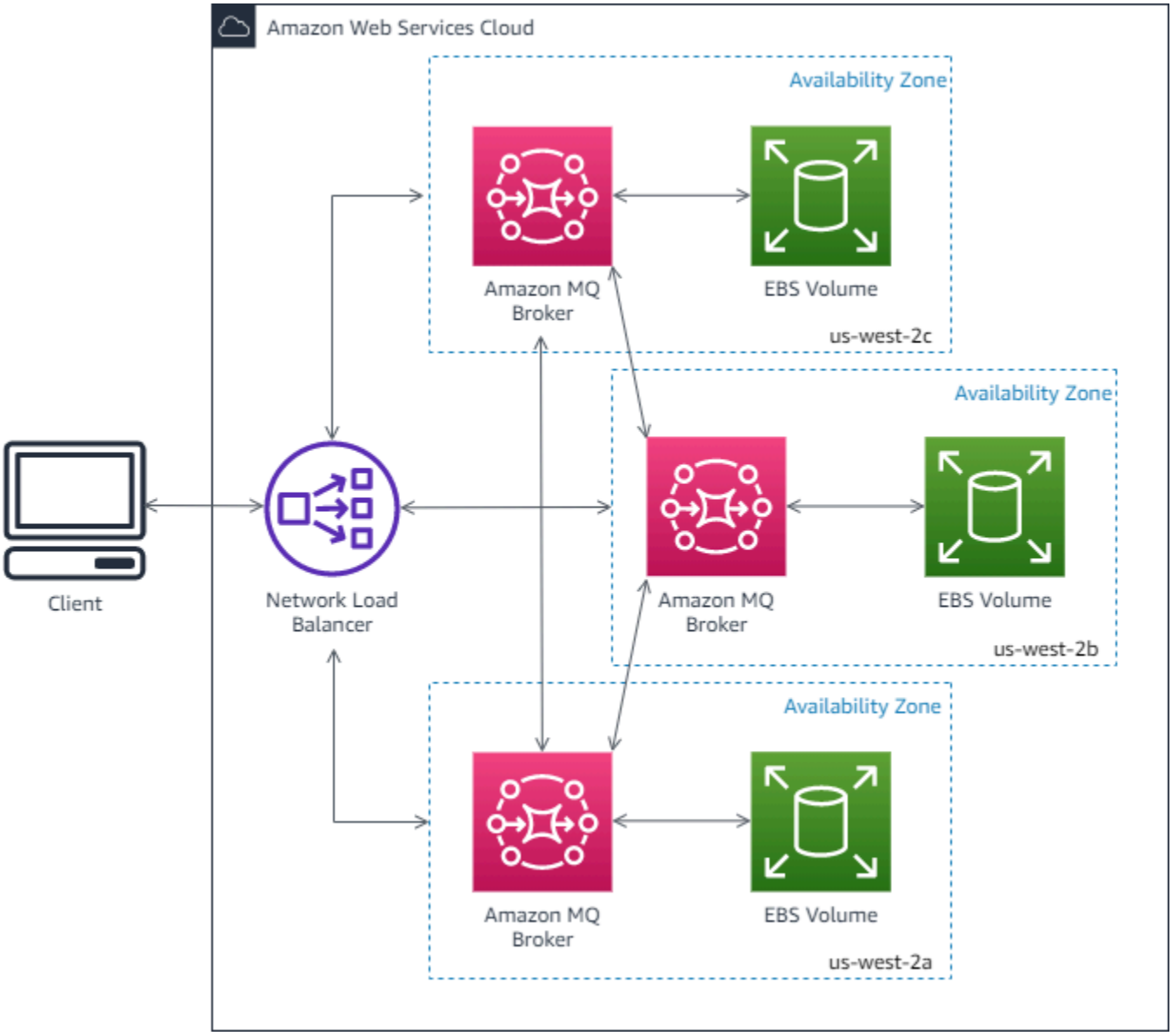
In einer Clusterbereitstellung verwaltet Amazon MQ automatisch Broker-Richtlinien, um die klassische Spiegelung über alle Knoten hinweg zu ermöglichen, wodurch eine hohe Verfügbarkeit (HA) sichergestellt wird. Jede gespiegelte Warteschlange besteht aus einemHaupt-Knoten und einen oder mehrereSpiegeln. Jede Warteschlange hat einen eigenen Hauptknoten. Alle Operationen für eine bestimmte Warteschlange werden zuerst auf den Hauptknoten der Warteschlange angewendet und dann an Spiegelungen weitergegeben. Amazon MQ erstellt eine Standard-Systemrichtlinie, die die`ha-mode aufallundha-sync-modeaufautomatic`. Dadurch wird sichergestellt, dass Daten auf alle Knoten im Cluster über verschiedene Availability Zones hinweg repliziert werden, um eine bessere Haltbarkeit zu gewährleisten.

> **Note**
>
> Während eines -Wartungsfensters wird die gesamte Wartung eines Clusters jeweils jeweils ein Knoten durchgeführt, wobei mindestens zwei laufende Knoten zu jeder Zeit beibehalten werden. Jedes Mal, wenn ein Knoten heruntergefahren wird, werden Clientverbindungen zu diesem Knoten getrennt und müssen wieder hergestellt werden. Sie müssen sicherstellen, dass der Clientcode so konzipiert ist, dass er automatisch wieder eine Verbindung mit dem Cluster herstellt. Weitere Informationen über den Wiederherstellungsprozess finden Sie unter the section called “Automatische Wiederherstellung nach Netzwerkausfällen”.

Weil Amazon MQ `ha-sync-mode: automatic` während eines Wartungsfensters synchronisiert, werden die Warteschlangen synchronisiert, wenn jeder Knoten dem Cluster wieder beitritt. Die Warteschlangen-Synchronisierung blockiert alle anderen Warteschlangen Sie können die Auswirkungen der Warteschlangensynchronisierung während Wartungsfenstern verringern, indem Sie Warteschlangen kurz halten.

Die Standardrichtlinie sollte nicht gelöscht werden. Wenn Sie diese Richtlinie löschen, erstellt Amazon MQ sie automatisch neu. Amazon MQ stellt außerdem sicher, dass HA-Eigenschaften auf alle anderen Richtlinien angewendet werden, die Sie für einen geclusterten Broker erstellen. Wenn Sie eine Richtlinie ohne die HA-Eigenschaften hinzufügen, fügt Amazon MQ diese für Sie hinzu. Wenn Sie eine Richtlinie mit unterschiedlichen Eigenschaften für hohe Verfügbarkeit hinzufügen, ersetzt Amazon MQ diese. Weitere Informationen zur klassischen Spiegelung von finden Sie unterKlassische gespiegelte Warteschlangen.

Das folgende Diagramm zeigt eine RabbitMQ-Cluster-Broker-Bereitstellung mit drei Knoten in drei Availability Zones (AZ), von denen jede über ein eigenes EBS Amazon-Volume und einen gemeinsamen Status verfügt. Amazon EBS bietet Speicher auf Blockebene, der für niedrige Latenz und hohen Durchsatz optimiert ist.



Amazon MQ für RabbitMQ-Brokerkonfigurationen

Eine Konfiguration enthält alle Einstellungen für Ihren RabbitMQ-Broker im Cuttlefish-Format. Sie können eine Konfiguration erstellen, bevor Sie Broker erstellen. Sie können die Konfiguration dann auf einen oder mehrere Broker anwenden.

Attribute

Eine Broker-Konfiguration verfügt über mehrere Attribute, z. B.:

- Einen Namen (`MyConfiguration`)

- Eine ID (`c-1234a5b6-78cd-901e-2fgh-3i45j6k178l9`)
- Ein Amazon-Ressourcenname (ARN) (`arn:aws:mq:us-east-2:123456789012:configuration:c-1234a5b6-78cd-901e-2fgh-3i45j6k178l9`)

Eine vollständige Liste der Konfigurationsattribute finden Sie im Folgenden in der Amazon MQ REST API MQ-Referenz:

- RESTVorgangs-ID: Konfiguration
- RESTVorgangs-ID: Konfigurationen

Eine vollständige Liste der Konfigurationsrevisions-Attribute finden Sie im folgenden Abschnitt:

- RESTVorgangs-ID: Revision der Konfiguration
- RESTVorgangs-ID: Konfigurationsrevisionen

Themen

- Erstellen und Anwenden von RabbitMQ-Broker-Konfigurationen
- Eine Konfigurationsrevision von Amazon MQ für RabbitMQ bearbeiten
- RabbitMQ-Konfigurationsrichtlinien

Erstellen und Anwenden von RabbitMQ-Broker-Konfigurationen

Eine Konfiguration enthält alle Einstellungen für Ihren RabbitMQ-Broker im Cuttlefish-Format. Sie können eine Konfiguration erstellen, bevor Sie Broker erstellen. Sie können die Konfiguration dann auf mindestens einen Broker anwenden.

Das folgenden Beispiele zeigen, wie Sie eine RabbitMQ-Broker-Konfiguration mithilfe der AWS Management Console erstellen und anwenden.

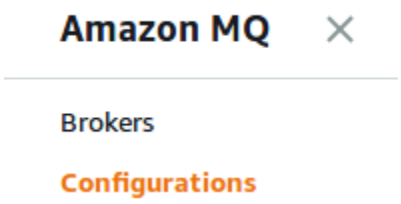
Themen

- Eine neue Konfiguration erstellen
- Erstellen einer neuen Konfigurationsversion
- Eine Konfigurationsrevision auf Ihren Broker anwenden

Eine neue Konfiguration erstellen

Um eine Konfiguration auf Ihren Broker anzuwenden, müssen Sie zuerst die Konfiguration erstellen.

1. Melden Sie sich bei der Amazon MQ-Konsole an.
2. Erweitern Sie den Navigationsbereich auf der linken Seite, und wählen Sie Configurations (Konfigurationen) aus.



3. Wählen Sie auf der Seite Configurations (Konfigurationen) die Option Create configuration (Konfiguration erstellen).
4. Geben Sie auf der Seite Create configuration (Konfiguration erstellen) im Abschnitt Details den Configuration name (Konfigurationsname) (z. B. `MyConfiguration`) ein und wählen Sie eine Broker-Engine-Version aus.

   Weitere Informationen zu RabbitMQ-Engine-Versionen, die von Amazon MQ für RabbitMQ unterstützt werden, finden Sie unter the section called “Versionsverwaltung.”.
5. Wählen Sie Create configuration (Konfiguration erstellen).

Erstellen einer neuen Konfigurationsversion

Nachdem Sie eine Konfiguration erstellt haben, müssen Sie die Konfiguration mithilfe einer Konfigurationsrevision bearbeiten.

1. Wählen Sie aus der Konfigurationsliste ***`MyConfiguration`***.

   > **Note**
   >
   > Die erste Revision der Konfiguration wird stets bei der Konfigurationserstellung durch Amazon MQ für Sie erstellt.

   Auf dem ***`MyConfiguration`***Auf der Seite werden der Broker-Engine-Typ und die Version angezeigt, die Ihre neue Konfigurationsrevision verwendet (z. B. RabbitMQ 3.xx.xx).

2. Auf der Registerkarte Konfigurationsdetails werden die Konfigurations-Revisionsnummer, die Beschreibung und die Broker-Konfiguration im Cuttlefish-Format angezeigt.

> **Note**
>
> Durch die Bearbeitung der aktuellen Konfiguration wird eine neue Konfigurationsversion erstellt.

3. Klicken Sie auf Konfiguration bearbeiten Nehmen Sie Änderungen an der Cuttlefish-Konfiguration vor.
4. Wählen Sie Save (Speichern) aus.

   DieSpeichern der Revisionwird angezeigt.
5. (Optional) Geben Sie `A description of the changes in this revision` ein.
6. Wählen Sie Save (Speichern) aus.

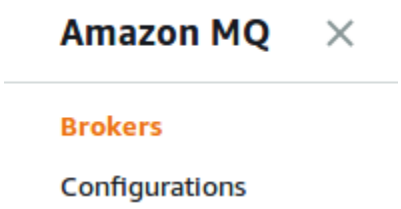
   Die neue Version der Konfiguration wird gespeichert.

> **Important**
>
> Das Vornehmen von Änderungen an einer Konfigurationnichtwenden Sie die Änderungen sofort an den Broker an. Um Ihre Änderungen zu übernehmen, müssen Sie auf den nächsten Wartungszeitraum warten oder den Broker neu starten.
> Derzeit ist es nicht möglich, eine Konfiguration zu löschen.

Eine Konfigurationsrevision auf Ihren Broker anwenden

Nachdem Sie die Konfigurationsrevision erstellt haben, können Sie die Konfigurationsrevision auf Ihren Broker anwenden.

1. Erweitern Sie den Navigationsbereich auf der linken Seite, und wählen Sie Broker aus.



2. Wählen Sie in der Brokerliste Ihren Broker aus (z. B. MyBroker) und klicken Sie dann auf Bearbeiten.
3. Auf der Seite Bearbeiten ***MyBroker***Wählen Sie auf der Seite im Abschnitt Konfiguration eine Konfiguration und eine Revision aus und klicken Sie dann auf Änderungen planen.
4. Wählen Sie im Abschnitt Schedule broker modifications (Broker-Änderungen planen) aus, ob die Änderungen During the next scheduled maintenance window (Im nächsten geplanten Wartungsfenster) oder Immediately (Sofort) angewendet werden sollen.

   > **Important**
   >
   > Ihr Broker ist offline, während er neu gestartet wird.

5. Wählen Sie Apply (Anwenden) aus.

   Ihre Konfigurationsversion wird zu der angegebenen Zeit auf Ihren Broker angewendet.

Eine Konfigurationsrevision von Amazon MQ für RabbitMQ bearbeiten

In den folgenden Anweisungen wird beschrieben, wie Sie eine Konfigurationsrevision für Ihren Broker bearbeiten.

1. Melden Sie sich bei der Amazon-MQ-Konsole an.
2. Wählen Sie in der Brokerliste Ihren Broker aus (z. B. MyBroker) und klicken Sie dann auf Bearbeiten.
3. Auf der ***MyBroker***Wählen Sie auf der Seite Bearbeiten aus.
4. Auf der Seite Bearbeiten ***MyBroker***Wählen Sie auf der Seite im Abschnitt Konfiguration eine Konfiguration und eine Revision aus und klicken Sie dann auf Bearbeiten.

   > **Note**
   >
   > Wenn Sie beim Erstellen eines Brokers eine Konfiguration auswählen, wird die erste Revision der Konfiguration stets bei der Konfigurationserstellung durch Amazon MQ für Sie erstellt.

   Auf der ***MyBroker***Auf der Seite werden der Broker-Engine-Typ und die Version angezeigt, die von der Konfiguration verwendet werden (z. B. RabbitMQ 3.xx.xx).

5. Auf der Registerkarte Konfigurationsdetails werden die Konfigurations-Revisionsnummer, die Beschreibung und die Broker-Konfiguration im Cuttlefish-Format angezeigt.

> **Note**
>
> Durch die Bearbeitung der aktuellen Konfiguration wird eine neue Konfigurationsversion erstellt.

6. Klicken Sie auf Konfiguration bearbeiten Nehmen Sie Änderungen an der Cuttlefish-Konfiguration vor.
7. Wählen Sie Save (Speichern) aus.

   DieSpeichern der Revisionwird angezeigt.
8. (Optional) Geben Sie `A description of the changes in this revision` ein.
9. Wählen Sie Save (Speichern) aus.

   Die neue Version der Konfiguration wird gespeichert.

> **Important**
>
> Das Vornehmen von Änderungen an einer Konfigurationnichtwenden Sie die Änderungen sofort an den Broker an. Um Ihre Änderungen zu übernehmen, müssen Sie auf den nächsten Wartungszeitraum warten oder den Broker neu starten.
> Derzeit ist es nicht möglich, eine Konfiguration zu löschen.

RabbitMQ-Konfigurationsrichtlinien

Amazon MQ für RabbitMQ unterstützt jetzt das Erstellen und Anwenden von Konfigurationen auf Ihren RabbitMQ-Broker. Die standardmäßige Operatorrichtlinie auf jedem virtuellen Host enthält die folgenden empfohlenen HA-Eigenschaften:

```
name: default_operator_policy_AWS_managed
pattern: .*
apply-to: all
priority: 0
definition: {
ha-mode: all
```

```
ha-sync-mode: automatic
}
```

Änderungen an den Operator-Richtlinien über AWS Management Console oder Management API sind standardmäßig nicht verfügbar. Sie können Änderungen aktivieren, indem Sie der Broker-Konfiguration die folgende Zeile hinzufügen:

```
management.restrictions.operator_policy_changes.disabled=false
```

Wenn Sie diese Änderung vornehmen, wird Ihnen dringend empfohlen, die HA-Eigenschaften in Ihre eigenen Betreiberrichtlinien aufzunehmen. Weitere Informationen zum Hinzufügen von Konfigurationen zu Ihrem Broker finden Sie unter Creating and applying broker configurations.

Konfiguration von HTTP Sicherheitsheadern

Die `secure.management.http.headers.enabled` Konfiguration ermöglicht die folgenden unveränderbaren HTTP Sicherheitsheader:

- X-Content-Type-Options: nosniff: verhindert, dass Browser Content Sniffing durchführen. Dabei handelt es sich um Algorithmen, die verwendet werden, um das Dateiformat von Websites abzuleiten.
- X-Frame-Options:DENY: verhindert, dass andere das Management-Plugin in einen Frame auf ihrer eigenen Website einbetten, um andere zu täuschen
- Strict-Transport-Security: max-age=47304000; includeSubDomains: erzwingt die Verwendung von Browsern HTTPS bei nachfolgenden Verbindungen zur Website und ihren Subdomains über einen längeren Zeitraum (1,5 Jahre).

Amazon MQ für RabbitMQ-Broker, die mit Versionen 3.10 und höher erstellt wurden, sind `secure.management.http.headers.enabled=true` standardmäßig auf eingestellt. `true` Sie können diese HTTP Sicherheitsheader aktivieren, indem Sie auf einstellen. `secure.management.http.headers.enabled=true` `true` Wenn Sie sich von diesen HTTP Sicherheitsheadern abmelden möchten, stellen Sie auf ein`secure.management.http.headers.enabled=true`. `false`

Quorum-Warteschlangen für RabbitMQ auf Amazon MQ

> **Important**
>
> Quorum-Warteschlangen sind nur für Makler auf Amazon MQ für RabbitMQ Version 3.13 und höher verfügbar.

Quorum-Warteschlangen sind replizierte Warteschlangen, die aus einem Leader (primäres Replikat) und Followern (andere Replikate) bestehen. Wenn der Leader nicht mehr verfügbar ist, wählt Quorum-Warteschlangen mithilfe des Raft-Konsensusalgorithmus mit Stimmenmehrheit einen neuen Leader-Knoten, und der vorherige Leiter wird zu einem Follower-Knoten im selben Cluster herabgestuft. Die verbleibenden Follower replizieren sich wie zuvor weiter. Da sich jeder Knoten in einer anderen Availability Zone befindet, wird die Nachrichtenzustellung mit dem neu gewählten Leader-Replikat in einer anderen Availability Zone fortgesetzt, wenn ein Knoten vorübergehend nicht verfügbar ist.

Quorumwarteschlangen sind nützlich für den Umgang mit giftigen Nachrichten, die entstehen, wenn eine Nachricht fehlschlägt und mehrfach in die Warteschlange gestellt wird.

Sie sollten Quorumwarteschlangen nicht verwenden, wenn Sie:

- Verwenden Sie vorübergehende Warteschlangen
- haben lange Warteschlangenrückstände
- priorisieren Sie niedrige Latenz

Um eine Quorumwarteschlange zu deklarieren, setzen Sie den Header `x-queue-type` auf. `quorum`

Themen

- Migration von klassischen Warteschlangen zu Quorum-Warteschlangen auf Amazon MQ für RabbitMQ
- Richtlinienkonfigurationen für Quorum-Warteschlangen für Amazon MQ für RabbitMQ
- Bewährte Methoden für Quorum-Warteschlangen für Amazon MQ für RabbitMQ

Migration von klassischen Warteschlangen zu Quorum-Warteschlangen auf Amazon MQ für RabbitMQ

Sie können Ihre klassischen gespiegelten Warteschlangen zu Quorum-Warteschlangen auf Amazon MQ-Brokern auf Version 3.13 oder höher migrieren, indem Sie einen neuen virtuellen Host auf demselben Cluster erstellen oder indem Sie vor Ort migrieren.

Option 1: Migration von klassischen gespiegelten Warteschlangen zu Quorum-Warteschlangen mit einem neuen virtuellen Host

Sie können Ihre klassischen gespiegelten Warteschlangen zu Quorum-Warteschlangen auf Amazon MQ-Brokern auf Version 3.13 oder höher migrieren, indem Sie einen neuen virtuellen Host auf demselben Cluster erstellen.

1. Erstellen Sie in Ihrem vorhandenen Cluster einen neuen virtuellen Host (vhost) mit dem Standard-Warteschlangentyp Quorum.
2. Erstellen Sie den Federation Plugin aus dem neuen Vhost, wobei der Vhost URI auf den alten Vhost verweist, und verwenden Sie dabei klassische gespiegelte Warteschlangen.
3. Verwenden Sie`rabbitmqadmin`, um die Definitionen aus dem alten Vhost in eine neue Datei zu exportieren. Sie müssen Änderungen an der Schemadatei vornehmen, damit sie mit Quorumwarteschlangen kompatibel ist. Eine vollständige Liste der Änderungen, die Sie an der Datei vornehmen müssen, finden Sie unter Definitionen verschieben in der RabbitMQ-Quorumwarteschlangendokumentation. Nachdem Sie die erforderlichen Änderungen an der Datei vorgenommen haben, importieren Sie die Definitionen erneut in den neuen Vhost.
4. Erstellen Sie eine neue Richtlinie im neuen Vhost. Empfehlungen zu Amazon MQ MQ-Richtlinienkonfigurationen für Quorumwarteschlangen finden Sie unter. Richtlinienkonfigurationen für Quorum-Warteschlangen für Amazon MQ für RabbitMQ Starten Sie dann den Verbund, den Sie zuvor erstellt haben, vom alten Vhost zum neuen Vhost.
5. Weisen Sie Verbraucher und Produzenten auf den neuen Vhost hin.
6. Konfigurieren Sie das Shovel-Plug-In so, dass alle verbleibenden Nachrichten übertragen werden. Sobald eine Warteschlange leer ist, löschen Sie den Shovel.

Migration von klassischen gespiegelten Warteschlangen zu Quorum-Warteschlangen ist bereits vorhanden

Sie können Ihre klassischen gespiegelten Warteschlangen zu Quorum-Warteschlangen auf Amazon MQ-Brokern mit Version 3.13 oder höher migrieren, indem Sie sie vor Ort migrieren.

1. Stoppt die Verbraucher und Produzenten.
2. Erstellen Sie eine neue temporäre Quorum-Warteschlange.
3. Konfigurieren Sie das Shovel-Plug-In so, dass alle Nachrichten aus der alten klassischen gespiegelten Warteschlange in die neue temporäre Quorumwarteschlange verschoben werden. Nachdem alle Nachrichten in die temporäre Quorum-Warteschlange verschoben wurden, löschen Sie Shovel.
4. Löschen Sie die klassische gespiegelte Quellwarteschlange. Erstellen Sie anschließend eine Quorumwarteschlange mit demselben Namen und denselben Bindungen wie die klassische gespiegelte Quellwarteschlange neu.
5. Erstellen Sie einen neuen Shovel, um die Nachrichten aus der temporären Quorum-Warteschlange in die neue Quorum-Warteschlange zu verschieben.

Richtlinienkonfigurationen für Quorum-Warteschlangen für Amazon MQ für RabbitMQ

Sie können spezifische Richtlinienkonfigurationen zu den Quorum-Warteschlangen für Ihren RabbitMQ-Broker auf Amazon MQ hinzufügen.

Wenn Sie eine Richtlinie für Quorumwarteschlangen erstellen, müssen Sie wie folgt vorgehen:

- Entfernen Sie alle Richtlinienattributeha, die mit,`ha-mode`, `ha-params` `ha-sync-mode` `ha-sync-batch-sizeha-promote-on-shutdown`, und beginnen. `ha-promote-on-failure`
- Entfernen Sie `queue-mode`.
- Ändern Sie den Überlauf, wenn er auf eingestellt ist `reject-publish-dlx`

> **Important**
>
> Amazon MQ for RabbitMQ wendet alle oder keines der Attribute innerhalb einer Richtlinie an. Sie können keine Richtlinie erstellen, die sowohl für klassische gespiegelte Warteschlangen als auch für Quorum-Warteschlangen gilt. Wenn Sie möchten, dass Ihre Richtlinie nur für

Quorumwarteschlangen gilt, müssen Sie auf einstellen. `--apply-to quorum_queues` Wenn Sie klassische gespiegelte Warteschlangen und Quorumwarteschlangen verwenden, müssen Sie eine separate Richtlinie mit`--apply-to`: `classic_queues` sowie eine Quorumwarteschlangenrichtlinie erstellen.

Sie müssen die `AWS-DEFAULT` Richtlinien nicht ändern, da sie automatisch den neuen Warteschlangentyp im Parameter „Gilt für“ übernehmen. Weitere Informationen zu Standardrichtlinien für Amazon MQ für RabbitMQ finden Sie unter. RabbitMQ configuration policies

Bewährte Methoden für Quorum-Warteschlangen für Amazon MQ für RabbitMQ

Wir empfehlen, die folgenden bewährten Methoden zu verwenden, um die Leistung bei der Arbeit mit Quorumwarteschlangen zu verbessern.

Umgang mit giftigen Nachrichten durch die Festlegung eines Zustelllimits

Verderbliche Nachrichten treten auf, wenn eine Nachricht fehlschlägt und mehrfach erneut zugestellt wird. Sie können ein Limit für die Nachrichtenzustellung festlegen, indem Sie das Argument `delivery-limit` policy verwenden, um Nachrichten zu verwerfen, die mehrfach erneut zugestellt werden. Wenn eine Nachricht öfter erneut zugestellt wird, als es das Zustellungslimit zulässt, wird die Nachricht dann von RabbitMQ gelöscht und gelöscht. Wenn du ein Zustellungslimit festlegst, wird die Nachricht an der Spitze der Warteschlange in die Warteschlange gestellt.

Nachrichtenpriorität für Quorum-Warteschlangen

Quorumwarteschlangen haben keine Nachrichtenpriorität. Wenn Sie Nachrichtenpriorität benötigen, müssen Sie mehrere Quorumwarteschlangen erstellen. Weitere Informationen zur Priorisierung von Nachrichten mit mehreren Quorumwarteschlangen finden Sie unter Nachrichtenpriorität in der RabbitMQ-Dokumentation.

Verwenden Sie den Standardreplikationsfaktor

Amazon MQ for RabbitMQ verwendet standardmäßig einen Replikationsfaktor von drei (3) Knoten für Cluster-Broker, die Quorum-Warteschlangen verwenden. Wenn Sie Änderungen an vornehmen`x-quorum-initial-group-size`, verwendet Amazon MQ wieder standardmäßig den Replikationsfaktor 3.

Fehlerbehebung RABBITMQ _ _ QUORUM _ QUEUES _ NOT SUPPORTED CURRENT _ON_ _ VERSION

Amazon MQ for RabbitMQ löst den Code für kritische erforderliche Aktionen aus, `RABBITMQ_QUORUM_QUEUES_NOT_SUPPORTED_ON_CURRENT_VERSION` wenn Sie versuchen, Quorum-Warteschlangen auf einer einzelnen Instance oder einem Cluster-Broker mit Version 3.12 und niedriger zu erstellen. Weitere Informationen zur Fehlerbehebung finden Sie unter. `RABBITMQ_QUORUM_QUEUES_NOT_SUPPORTED_ON_CURRENT_VERSION` Amazon MQ für RabbitMQ: Alarm für Quorum-Warteschlangen

RabbitMQ-Tutorials

Die folgenden Tutorials zeigen, wie Sie RabbitMQ in Amazon MQ konfigurieren und verwenden. Weitere Informationen zum Arbeiten mit unterstützten Clientbibliotheken in einer Vielzahl von Programmiersprachen wie Node.js, Python, .NET und mehr finden Sie unterRabbitMQ-TutorialsimHandbuch „RabbitMQ“.

Themen

- Bearbeiten von Broker-Einstellungen
- Verwenden von Python Pika mit Amazon MQ for RabbitMQ
- Auflösen der Synchronisierung von RabbitMQ angehaltener Warteschlangensynchronisierung

Bearbeiten von Broker-Einstellungen

Sie können Ihre Brokereinstellungen bearbeiten, z. B. CloudWatch Protokolle aktivieren oder deaktivieren, indem Sie dieAWS Management Console.

RabbitMQ-Broker-Optionen bearbeiten

1. Melden Sie sich bei der Amazon MQ-Konsole an.
2. Wählen Sie in der Brokerliste Ihren Broker aus (z. B.MyBroker) und danach wählen Sie .Bearbeiten.
3. Wählen Sie auf der Seite ***MyBroker*** bearbeiten im Abschnitt Spezifikationen eine Broker-Engine-Version oder einen Broker-Instance-Typ.
4. Im Abschnitt CloudWatch Logs, klicken Sie auf die Umschaltfläche, um allgemeine Protokolle zu aktivieren oder zu deaktivieren. Keine weiteren erforderlichen Schritte.

> **Note**
>
> - Für RabbitMQ-Broker verwendet Amazon MQ automatisch eine Service-Linked Role (SLR), um allgemeine Protokolle in CloudWatch zu veröffentlichen. Weitere Informationen finden Sie unter the section called “Verwenden von servicegebundenen Rollen”
> - Amazon MQ unterstützt keine Überwachungsprotokollierung für RabbitMQ-Broker.

5. Konfigurieren Sie im Abschnitt Wartung den Wartungszeitplan für Ihren Broker:

   Um Upgrades auf neue Versionen Ihres Brokers vorzunehmen, wenn sie von AWS veröffentlicht werden, wählen Sie Automatische Upgrades von Unterversionen aktivieren. Automatische Upgrades werden während der-Wartungsfensterdefiniert durch den Wochentag, die Tageszeit (im 24-Stunden-Format) und die Zeitzone (standardmäßig UTC).

6. Wählen Sie Änderungen einplanen.

> **Note**
>
> Wenn Sie nur Automatische kleinere Aktualisierungen aktivieren wählen, wechselt die Schaltfläche zu Speichern, da kein Neustart des Brokers erforderlich ist.

Ihre Einstellungen werden zu der angegebenen Zeit auf Ihren Broker angewendet.

Verwenden von Python Pika mit Amazon MQ for RabbitMQ

Das folgende Tutorial zeigt, wie Sie einen Python-Pika-Client mit TLS einrichten können, der für die Verbindung zu einem Amazon-MQ-for-RabbitMQ-Broker konfiguriert ist. Pika ist eine Python-Implementierung des AMQP-0-9-1-Protokolls für RabbitMQ. Dieses Tutorial führt Sie durch die Installation von Pika, das Deklarieren einer Warteschlange, das Einrichten eines Herausgebers für das Senden von Nachrichten an den Standardaustausch des Brokers und das Einrichten eines Verbrauchers, der Nachrichten aus der Warteschlange erhält.

Themen

- Voraussetzungen

Voraussetzungen

Um die Schritte dieses Tutorials auszuführen, benötigen Sie Folgendes:

- Einen Amazon-MQ-for-RabbitMQ-Broker. Weitere Informationen finden Sie unter Erstellen eines Amazon-MQ-for-RabbitMQ-Brokers.
- Python 3 für Ihr Betriebssystem installieren.
- Pika mithilfe von Python `pip` installiert. Öffnen Sie zum Installieren von Pika ein neues Terminalfenster und führen Sie Folgendes aus.

```
$ python3 -m pip install pika
```

Berechtigungen

Für dieses Tutorial benötigen Sie mindestens einen Amazon-MQ-for-RabbitMQ-Brokerbenutzer mit der Berechtigung, an einen Vhost zu schreiben und von ihm zu lesen. Die folgende Tabelle enthält die erforderlichen Mindestberechtigungen als reguläre Ausdrücksmuster (regexp).

Tags (Markierungen)	Konfigurieren von regexp	REGEXP-Schreiben	Lesen Sie regexp
none		`.*`	`.*`

Die aufgelisteten Benutzerberechtigungen bieten dem Benutzer nur Lese- und Schreibberechtigungen, ohne Zugriff auf das Management-Plug-In zu gewähren, um Verwaltungsvorgänge für den Broker auszuführen. Sie können Berechtigungen weiter einschränken, indem Sie regexp-Muster bereitstellen, die den Zugriff des Benutzers auf bestimmte Warteschlangen

einschränken. Zum Beispiel, wenn Sie das Lese-regexp-Muster auf `^[hello world].*` ändern, hat der Benutzer nur die Berechtigung, aus Warteschlangen zu lesen, die mit `hello world` starten.

Weitere Informationen zum Erstellen von RabbitMQ-Benutzern und zum Verwalten von Benutzer-Tags und -Berechtigungen finden Sie unterAmazon MQ für RabbitMQ-Broker-Benutzer.

Schritt eins: Erstellen Sie einen einfachen Python-Pika-Client

Um eine Python-Pika-Client-Basisklasse zu erstellen, die einen Konstruktor definiert und den SSL-Kontext bereitstellt, der für die TLS-Konfiguration erforderlich ist, wenn Sie mit einem Amazon-MQ-for-RabbitMQ-Broker interagieren, machen Sie folgendes.

1. Öffnen Sie ein neues Terminalfenster, erstellen Sie ein neues Verzeichnis für Ihr Projekt und navigieren Sie zum Verzeichnis.

```
$ mkdir pika-tutorial
$ cd pika-tutorial
```

2. Erstellen Sie eine neue Datei, `basicClient.py`, die folgenden Python-Code enthält.

```
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()
```

Sie können jetzt zusätzliche Klassen für Ihren Herausgeber und Verbraucher definieren, die von `BasicPikaClient` erben.

Schritt zwei: Erstellen Sie einen Herausgeber und senden Sie eine Nachricht

Gehen Sie wie folgt vor, um einen Herausgeber zu erstellen, der eine Warteschlange deklariert und eine einzelne Nachricht sendet.

1. Kopieren Sie den Inhalt des folgenden Codebeispiels und speichern Sie es lokal als `publisher.py` im selben Verzeichnis, das Sie im vorherigen Schritt erstellt haben.

```
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()
```

Die `BasicMessageSender`-Klasse erbt von `BasicPikaClient` und implementiert zusätzliche Methoden zum Deklarieren einer Warteschlange, zum Senden einer Nachricht an die Warteschlange und zum Schließen von Verbindungen. Das Codebeispiel leitet eine Nachricht an den Standardaustausch weiter, wobei ein Routing-Schlüssel dem Namen der Warteschlange entspricht.

2. Unter `if __name__ == "__main__":`, ersetzen Sie die Parameter, die an die `BasicMessageSender`-constructor-Anweisung weitergegeben werden mit den folgenden Informationen.

   - **<broker-id>** - Die eindeutige ID, die Amazon MQ für die Broker-Instance generiert. Sie können die ID von Ihrem Broker ARN analysieren. Beispielsweise angesichts der folgenden ARN, `arn:aws:mq:us-east-2:123456789012:broker:MyBroker:b-1234a5b6-78cd-901e-2fgh-3i45j6k178l9`, wäre die Broker-ID `b-1234a5b6-78cd-901e-2fgh-3i45j6k178l9`.
   - **<username>** – Der Benutzername für einen Broker-Benutzer mit ausreichenden Berechtigungen zum Schreiben von Nachrichten an den Broker.
   - **<password>** – Das Passwort für einen Broker-Benutzer mit ausreichenden Berechtigungen zum Schreiben von Nachrichten an den Broker.
   - **<region>** – DieAWS-Region, in der Sie Ihren Amazon-MQ-for-RabbitMQ-Broker erstellt haben. Zum Beispiel `us-west-2`.

3. Führen Sie den folgenden Befehl im selben Verzeichnis aus, in dem Sie `publisher.py` erstellt haben.

```
$ python3 publisher.py
```

Wenn der Code erfolgreich ausgeführt wird, wird die folgende Meldung in Ihrem Terminalfenster angezeigt.

```
Trying to declare queue(hello world queue)...
Sent message. Exchange: , Routing Key: hello world queue, Body: b'Hello World!'
```

Schritt drei: Erstellen Sie einen Verbraucher und erhalten Sie eine Nachricht

Gehen Sie wie folgt vor, um einen Verbraucher zu erstellen, der eine einzelne Nachricht aus der Warteschlange erhält.

1. Kopieren Sie den Inhalt des folgenden Codebeispiels und speichern Sie es lokal als `consumer.py` im selben Verzeichnis.

```
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()
```

Ähnlich wie bei dem Herausgeber, den Sie im vorherigen Schritt erstellt haben, erbt `BasicMessageReciever` von `BasicPikaClient` und implementiert zusätzliche Methoden zum Empfangen einer einzelnen Nachricht und zum Schließen von Verbindungen.

2. In der `if __name__ == "__main__":`-Anweisung, ersetzen Sie die Parameter, die an den `BasicMessageReciever`-Constructor weitergegeben werden mit Ihren Informationen.
3. Führen Sie den folgenden Befehl in Ihrem Projektverzeichnis aus.

```
$ python3 consumer.py
```

Wenn der Code erfolgreich ausgeführt wird, werden der Nachrichtentext und die Header einschließlich des Routing-Schlüssels in Ihrem Terminalfenster angezeigt.

```
<Basic.GetOk(['delivery_tag=1', 'exchange=', 'message_count=0',
 'redelivered=False', 'routing_key=hello world queue'])> <BasicProperties> b'Hello
 World!'
```

Schritt vier: (Optional) Richten Sie eine Ereignisschleife ein und konsumieren Sie Nachrichten

Um mehrere Nachrichten aus einer Warteschlange zu konsumieren, verwenden Sie Pikas basic_consume-Methode und eine Callback-Funktion wie nachfolgend dargestellt

1. In `consumer.py`, fügen Sie die folgende Methodendefinition zur `BasicMessageReceiver`-Klasse hinzu.

```
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. In `consumer.py`, unter `if __name__ == "__main__":`, rufen Sie die `consume_messages`-Methode auf, die Sie im vorherigen Schritt definiert haben.

```
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. Führen Sie `consumer.py` erneut aus, und falls dies erfolgreich ist, werden die Nachrichten in der Warteschlange in Ihrem Terminalfenster angezeigt.

```
[*] Waiting for messages. To exit press CTRL+C
[x] Received b'Hello World!'
[x] Received b'Hello World!'
...
```

Als nächstes

- Weitere Informationen zu anderen unterstützten RabbitMQ-Clientbibliotheken finden Sie in der RabbitMQ-Client-Dokumentation auf der RabbitMQ-Website.

Auflösen der Synchronisierung von RabbitMQ angehaltener Warteschlangensynchronisierung

In einem Amazon MQ für RabbitMQ Cluster-Bereitstellung, werden Nachrichten, die in jeder Warteschlange veröffentlicht werden, über drei Broker-Knoten repliziert. Diese Replikation, bezeichnet als Spiegelung, bietet Hochverfügbarkeit (HA) für RabbitMQ-Broker. Warteschlangen in einer Clusterbereitstellung bestehen aus einemHauptReplikat auf einem Knoten und einem oder mehrerenMirror. Jeder Vorgang, der auf eine gespiegelte Warteschlange angewendet wird, einschließlich der Warteschlange, wird zuerst auf die Hauptwarteschlange angewendet und dann über ihre Spiegelungen repliziert.

Betrachten Sie beispielsweise eine gespiegelte Warteschlange, die über drei Knoten repliziert wird: den Hauptknoten (`main`) und zwei Spiegeln (`mirror-1`und`mirror-2`) enthalten. Wenn alle Nachrichten in dieser gespiegelten Warteschlange erfolgreich an alle Spiegelungen weitergegeben werden, wird die Warteschlange synchronisiert. Wenn ein Knoten (`mirror-1`) für ein Zeitintervall nicht verfügbar ist, ist die Warteschlange noch funktionsfähig und kann weiterhin Nachrichten in die Warteschlange einlegen. Damit die Warteschlange synchronisiert werden kann, werden Nachrichten, die in`main`WHILE`mirror-1`nicht verfügbar ist, muss repliziert werden`mirror-1`.

Weitere Informationen zum Spiegelung finden Sie unter Klassische gespiegelte Warteschlangen auf der RabbitMQ-Website.

Wartung und Warteschlangensynchronisierung

Während Wartungsfenstern führt Amazon MQ alle Wartungsarbeiten jeweils einen Knoten aus, um sicherzustellen, dass der Broker betriebsbereit bleibt. Daher müssen Warteschlangen möglicherweise synchronisiert werden, wenn jeder Knoten den Vorgang fortsetzt. Während der Synchronisierung werden Nachrichten, die auf Spiegelungen repliziert werden müssen, vom entsprechenden Amazon Elastic Block Store (Amazon EBS) -Volume in den Speicher geladen, um in Batches verarbeitet zu werden. Durch die Verarbeitung von Nachrichten in Batches können Warteschlangen schneller synchronisiert werden.

Wenn Warteschlangen kurz gehalten werden und Nachrichten klein sind, werden die Warteschlangen erfolgreich synchronisiert und wie erwartet fortgesetzt. Wenn sich die Datenmenge in einem Batch jedoch dem Speicherlimit des Knotens nähert, löst der Knoten einen Alarm mit hohem Speicher aus, der die Warteschlangen-Synchronisierung pausiert. Sie können die Speicherauslastung bestätigen, indem Sie die`RabbitMemUsed`und`RabbitMqMemLimit` Metriken von Brokerknoten in CloudWatch. Die Synchronisierung kann erst abgeschlossen werden, wenn Nachrichten verbraucht oder gelöscht oder die Anzahl der Nachrichten im Stapel reduziert wird.

> **Note**
>
> Die Reduzierung der Stapelgröße der Warteschlangensynchronisierung kann zu einer höheren Anzahl von Replikationstransaktionen führen.

Um eine angehaltene Warteschlangensynchronisierung aufzulösen, führen Sie die Schritte in diesem Lernprogramm aus, in dem veranschaulicht wird, wie eine `ha-sync-batch-size`-Richtlinie angewendet wird, und starten Sie die Warteschlangen-Synchronisierung neu.

Themen

Voraussetzungen

Für dieses Tutorial benötigen Sie einen Amazon MQ for RabbitMQ Broker Benutzer mit Administratorberechtigungen. Sie können den Administratorbenutzer verwenden, der beim ersten Erstellen des Brokers erstellt wurde, oder einen anderen Benutzer, den Sie später erstellt haben. Die folgende Tabelle enthält die erforderlichen Administratorbenutzer-Tag und Berechtigungen als reguläre Ausdrücke (regexp) Muster.

Tags (Markierungen)	Lesen Sie regexp	Konfigurieren von regexp	REGEXP-Schreiben
`administrator`	`.*`	`.*`	`.*`

Weitere Informationen zum Erstellen von RabbitMQ-Benutzern und zum Verwalten von Benutzer-Tags und -Berechtigungen finden Sie unterAmazon MQ für RabbitMQ-Broker-Benutzer.

Schritt 1: Wenden Sie eine **`ha-sync-batch-size`** Richtlinie an

Die folgenden Verfahren veranschaulichen das Hinzufügen einer Richtlinie, die für alle Warteschlangen gilt, die auf dem Broker erstellt wurden. Sie können die RabbitMQ-Webkonsole oder die RabbitMQ-Management-API verwenden. Weitere Informationen finden Sie unter Management-Plugin auf der RabbitMQ-Website.

So wenden Sie eine **`ha-sync-batch-size`**-Richtlinie mit der RabbitMQ-Webkonsole an

1. Melden Sie sich bei der Amazon MQ-Konsole an.
2. Wählen Sie im linken Navigationsbereich die Option Broker aus.
3. Wählen Sie in der Broker-Liste den Namen des Brokers aus, auf den Sie die neue Richtlinie anwenden möchten.
4. Auf der Seite des Brokers im-Verbindungen, wählen Sie im Bereich die OptionRabbitMQ WebkonsoleURL. Die RabbitMQ-Webkonsole wird in einer neuen Browserregisterkarte oder -fenster geöffnet.
5. Melden Sie sich mit Ihren Broker-Administratoranmeldeinformationen bei der RabbitMQ-Webkonsole an.
6. Wählen Sie in der RabbitMQ-Webkonsole oben auf der Seite die OptionAdmin.
7. Klicken Sie auf derAdminWählen Sie im rechten Navigationsbereich die OptionRichtlinien.
8. Klicken Sie auf derRichtlinienkönnen Sie eine Liste der aktuellen Broker-Benutzerrichtlinien sehen. UnterBenutzerrichtlinienErweitern Sie mitSo fügen/aktualisieren Sie eine Richtlinie.

   > **Note**
   >
   > Standardmäßig werden Amazon MQ für RabbitMQ-Cluster mit einer anfänglichen Broker-Richtlinie namens`ha-all-AWS-OWNED-DO-NOT-DELETE`. Amazon MQ verwaltet diese Richtlinie, um sicherzustellen, dass jede Warteschlange im Broker auf alle drei Knoten repliziert wird und dass Warteschlangen automatisch synchronisiert werden.

9. Um eine neue Broker-Richtlinie zu erstellen, gehen Sie unter Eine Richtlinie hinzufügen/aktualisieren wie folgt vor:

   a. Geben Sie unter Name einen Namen für Ihre Richtlinie ein, z. B. **`batch-size-policy`**.

   b. Für Pattern geben Sie das regexp-Muster ein**`.*`**, damit die Richtlinie mit allen Warteschlangen auf dem Broker übereinstimmt.

c. FürÜbernehmen von, wählen Sie Tauschen von Warteschlangen aus der Dropdown-Liste.

d. FürPriorität, geben Sie eine Ganzzahl ein, die größer ist als alle anderen Richtlinien, die auf den vhost angewendet werden. Sie können jederzeit genau einen Satz von Richtliniendefinitionen auf RabbitMQ-Warteschlangen und -Austauschvorgänge anwenden. RabbitMQ wählt die Matching-Policy mit dem höchsten Prioritätswert . Weitere Informationen zu Richtlinienprioritäten und zum Kombinieren von Richtlinien finden Sie unter Richtlinien in der Dokumentation zu RabbitMQ Server.

e. FürDefinition, fügen Sie die folgenden Schlüssel/Wert-Paare hinzu:

- **`ha-sync-batch-size`**=*`100`*USD Wählen Sie Nummer aus der Dropdown-Liste.

  > **Note**
  >
  > Möglicherweise müssen Sie den Wert von `ha-sync-batch-size` basierend auf der Anzahl und Größe der nicht synchronisierten Nachrichten in Ihren Warteschlangen anpassen.

- **`ha-mode=all`**. Klicken Sie aufZeichenfolgeaus der Dropdown-Liste.

  > **Important**
  >
  > Die `ha-mode`-Definition ist für alle HA-bezogenen Richtlinien erforderlich. Das Auslassen führt zu einem Validierungsfehler.

- **`ha-sync-mode=automatic`**. Klicken Sie aufZeichenfolgeaus der Dropdown-Liste.

  > **Note**
  >
  > Die `ha-sync-mode`-Definition ist für alle benutzerdefinierten Richtlinien erforderlich. Wenn sie nicht angegeben wird, hängt Amazon MQ die Definition automatisch an.

f. Wählen Sie Richtlinie aktualisieren.

10. Vergewissern Sie sich, dass die neue Richtlinie in der Liste derBenutzerrichtlinien erscheint.

So verwenden Sie eine**`ha-sync-batch-size`**-Richtlinie mit der RabbitMQ-Verwaltungs-API

1. Melden Sie sich bei der Amazon MQ-Konsole an.
2. Wählen Sie im linken Navigationsbereich die Option Broker aus.
3. Wählen Sie in der Broker-Liste den Namen des Brokers aus, auf den Sie die neue Richtlinie anwenden möchten.
4. Auf der Seite des Brokers im -Verbindungen-Abschnitt, notieren Sie sich dieRabbitMQ WebkonsoleURL. Dies ist der Broker-Endpunkt, den Sie in einer HTTP-Anforderung verwenden.
5. Öffnen Sie ein neues Terminal- oder Befehlszeilenfenster Ihrer Wahl.
6. Um eine neue Broker-Richtlinie zu erstellen, geben Sie Folgendes ein`curl`-Befehl. Dieser Befehl nimmt an, dass eine Warteschlange auf der`/`vhost, der als`%2F` encodiert ist.

   > **Note**
   >
   > Ersetzen Sie den *`Benutzernamen`* und das *`Passwort`* durch Ihre Broker-Administratoranmeldeinformationen. Möglicherweise müssen Sie den Wert von `ha-sync-batch-size`(*`100`*) basierend auf der Anzahl und Größe der nicht synchronisierten Nachrichten in Ihren Warteschlangen anpassen. Ersetzen Sie den Broker-Endpunkt durch die URL, die Sie zuvor notiert haben.

   ```
   curl -i -u username:password -H "content-type:application/json" -XPUT \
   -d '{"pattern":".*", "priority":1, "definition":{"ha-sync-batch-size":100, "ha-
   mode":"all", "ha-sync-mode":"automatic"}}' \
   https://b-589c045f-f8ln-4ab0-a89c-co62e1c32ef8.mq.us-west-2.amazonaws.com/api/
   policies/%2F/batch-size-policy
   ```

7. Um zu bestätigen, dass die neue Richtlinie den Benutzerrichtlinien Ihres Brokers hinzugefügt wird, geben Sie folgen `curl`-Befehl, um alle Broker-Richtlinien aufzulisten.

   ```
   curl -i -u username:password https://b-589c045f-f8ln-4ab0-a89c-co62e1c32ef8.mq.us-
   west-2.amazonaws.com/api/policies
   ```

Schritt 2: Starten Sie die Warteschlangen-Synchronisierung

Nach dem Anwenden einer neuen`ha-sync-batch-size`-Richtlinie an Ihren Broker, starten Sie die Warteschlangen-Synchronisierung neu.

So starten Sie die Warteschlangensynchronisierung mithilfe der RabbitMQ-Webkonsole neu

> **Note**
>
> Informationen zum Öffnen der RabbitMQ-Webkonsole finden Sie in den vorherigen Anweisungen in Schritt 1 dieses Lernprogramms.

1. Wählen Sie in der RabbitMQ-Webkonsole oben auf der Seite die OptionQueues (Warteschlangen).
2. Klicken Sie auf die Seite Queues (Warteschlangen), und suchen Sie Ihre angehaltene Warteschlange unterAlle Warteschlangen. In der-Funktionen-Spalte, sollte die Warteschlange den Namen der neuen Richtlinie auflisten, die Sie erstellt haben (z. B.`batch-size-policy`).
3. Um den Synchronisierungsprozess mit einer reduzierten Stapelgröße neu zu starten, wählen SieSynchronisation neu starten.

> **Note**
>
> Wenn die Synchronisation angehalten wird und nicht erfolgreich abgeschlossen wird, versuchen Sie, den `ha-sync-batch-size`-Wert zu reduzieren und starten Sie die Warteschlangen-Synchronisierung erneut.

Nächste Schritte

- Sobald Ihre Warteschlange erfolgreich synchronisiert wurde, können Sie die Speichermenge überwachen, die Ihre RabbitMQ-Knoten verwenden, indem Sie die Amazon CloudWatch Metrik`RabbitMQMemUsed`. Sie können auch die `RabbitMQMemLimit`-Metrik, um das Speicherlimit eines Knotens zu überwachen. Weitere Informationen finden Sie unter Zugreifen auf CloudWatch Metriken für Amazon MQ und Verfügbare CloudWatch Metriken für Amazon MQ für RabbitMQ-Broker.

- Um eine angehaltene Warteschlangensynchronisierung zu verhindern, empfehlen wir, Warteschlangen kurz zu halten und Nachrichten zu verarbeiten. Für Workloads mit größeren Nachrichtengrößen empfehlen wir außerdem, Ihren Broker-Instance-Typ auf eine größere Instance-Größe mit mehr Speicher zu aktualisieren. Weitere Informationen zu Broker-Instance-Typen und zum Bearbeiten von Broker-Voreinstellungen finden Sie unterInstance-Typen von Amazon MQ für RabbitMQundBearbeiten von Broker-Einstellungen.
- Wenn Sie einen neuen Amazon MQ für RabbitMQ Broker erstellen, wendet Amazon MQ eine Reihe von Standardrichtlinien und virtuellen Host-Limits an, um die Broker-Performance zu optimieren. Wenn Ihr Broker nicht über die empfohlenen Standardrichtlinien und -beschränkungen verfügt, empfehlen wir, diese selbst zu erstellen. Weitere Informationen zum Erstellen von Standardrichtlinien und Vhost-Grenzwerten finden Sie unterthe section called “Standardeinstellungen für Broker”.

Zugehörige Ressourcen

- UpdateBrokerInput— Verwenden Sie diese Broker-Eigenschaft, um einen Broker-Instance-Typ mithilfe der Amazon MQ-API zu aktualisieren.
- Parameter und Richtlinien(RabbitMQ Server Documentation) — Erfahren Sie mehr über RabbitMQ-Parameter und -Richtlinien auf der RabbitMQ-Website.
- RabbitMQ-Management HTTP-API— Erfahren Sie mehr über die RabbitMQ-Management-API.

Verwalten von Amazon-MQ-für-RabbitMQ-Engine-Versionen

RabbitMQ organisiert Versionsnummern gemäß der semantischen Versioning-Spezifikation als `X.Y.Z`. `X`Bezeichnet in Implementierungen von Amazon MQ für RabbitMQ die Hauptversion, `Y` steht für die Nebenversion und `Z` gibt die Patch-Versionsnummer an. Amazon MQ betrachtet eine Versionsänderung als Hauptversionsänderung, wenn sich die Hauptversionsnummern ändern. Beispielsweise wird ein Upgrade von Version 3.13 auf 4.0 als Hauptversions-Upgrade betrachtet. Eine Versionsänderung gilt als geringfügig, wenn sich nur die Versionsnummer der Nebenversion oder des Patches ändert. Zum Beispiel ein Upgrade von Version 3. 1.1 2.8 auf 3. 1.2 1.3 wird als geringfügiges Versionsupgrade betrachtet.

Amazon MQ for RabbitMQ empfiehlt allen Brokern, die neueste unterstützte Nebenversion zu verwenden. Anweisungen zum Upgrade Ihrer Broker-Engine-Version finden Sie unter Upgrade einer Amazon MQ-Broker-Engine-Version.

> **Important**
>
> Amazon MQ unterstützt keine Streams. Wenn Sie dieses Feature-Flag aktivieren und einen Stream erstellen, führt dies zu Datenverlust.
> Amazon MQ unterstützt die in JSON RabbitMQ 3.9 eingeführte strukturierte Anmeldung nicht

Unterstützte Engine-Versionen auf Amazon MQ für RabbitMQ

Der Support-Kalender für die Amazon MQ MQ-Version gibt an, wann der Support für eine Broker-Engine-Version endet. Wenn für eine Version der Support ausläuft, aktualisiert Amazon MQ alle Broker dieser Version automatisch auf die nächste unterstützte Version. Dieses Upgrade findet während der geplanten Wartungsfenster Ihres Brokers innerhalb von 45 Tagen nach dem end-of-support Datum statt.

Amazon MQ informiert Sie mindestens 90 Tage im Voraus, bevor der Support für eine Version endet. Wir empfehlen, Ihren Broker vor diesem end-of-support Datum zu aktualisieren, um Störungen zu vermeiden. Darüber hinaus können Sie innerhalb von 30 Tagen nach Ablauf des Supports keine neuen Broker für Versionen erstellen, für die das Ende des Supports geplant ist.

RabbitMQ-Version	Ende des Supports bei Amazon MQ
3.13 (empfohlen)	
3.12	
3,11	
3,10	15. Oktober 2024
3.9	16. September 2024

Wenn Sie einen neuen Amazon MQ für RabbitMQ Broker erstellen, können Sie jede unterstützte Version der RabbitMQ Engine angeben. Wenn Sie bei der Erstellung eines Brokers keine Engine-Versionsnummer angeben, verwendet Amazon MQ automatisch standardmäßig die neueste Engine-Versionsnummer.

Upgrades der Engine-Version

Sie können Ihren Broker jederzeit manuell auf die nächste unterstützte Haupt- oder Nebenversion aktualisieren. Wenn Sie automatische Upgrades für Nebenversionen aktivieren, aktualisiert Amazon MQ Ihren Broker während des Wartungsfensters auf die neueste unterstützte Patch-Version.

Weitere Informationen zur manuellen Aktualisierung Ihres Brokers finden Sie unterthe section called “Upgrade der Engine-Version”.

Für alle Broker, die Engine-Version 3.13 und höher verwenden, verwaltet Amazon MQ während des Wartungsfensters Upgrades auf die neueste unterstützte Patch-Version.

> **Important**
>
> RabbitMQ erlaubt nur inkrementelle Versionsaktualisierungen (z. B. von 3.9.x auf 3.10.x). Sie können bei der Aktualisierung keine Nebenversionen überspringen (z. B.: 3.8.x auf 3.11.x).

Single-Instance-Broker sind während des Neustarts offline. Bei Cluster-Brokern müssen die gespiegelten Warteschlangen beim Neustart synchronisiert werden. Bei längeren Warteschlangen kann die Warteschlangensynchronisierung länger dauern. Während der Warteschlangensynchronisierung ist die Warteschlange für Verbraucher und Produzenten nicht verfügbar. Wenn die Warteschlangensynchronisierung abgeschlossen ist, ist der Broker wieder verfügbar. Um die Auswirkungen so gering wie möglich zu halten, empfehlen wir, das Upgrade während einer Zeit mit geringem Datenverkehr durchzuführen. Weitere Informationen zu bewährten Methoden für Versionsupgrades finden Sie unterBest Practices für Amazon MQ for RabbitMQ.

Unterstützte Engine-Versionen auflisten

Mithilfe des `describe-broker-instance-options` AWS CLI Befehls können Sie alle unterstützten Neben- und Hauptversionen der Engine auflisten.

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

Um die Ergebnisse nach Engine und Instance-Typ zu filtern, verwenden Sie die `--engine-type`- und `--host-instance-type`-Optionen wie im Folgenden gezeigt.

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

Um die Ergebnisse beispielsweise nach RabbitMQ und `mq.m5.large` Instanztyp zu filtern, ersetzen Sie *`engine-type`* mit und `RABBITMQ` *`instance-type`* mit`mq.m5.large`.

Best Practices für Amazon MQ for RabbitMQ

Verwenden Sie dies als Referenz, um schnell Empfehlungen zur Maximierung der Leistung und Minimierung der Durchsatzkosten bei der Arbeit mit RabbitMQ-Brokern auf Amazon MQ zu finden.

> **Important**
>
> Derzeit unterstützt Amazon MQ keine Streams oder die Verwendung der strukturierten Anmeldung, die in RabbitMQ JSON 3.9.x eingeführt wurde.

> **Important**
>
> Amazon MQ for RabbitMQ unterstützt den Benutzernamen „guest“ nicht und löscht das Standard-Gastkonto, wenn Sie einen neuen Broker erstellen. Amazon MQ löscht außerdem regelmäßig alle vom Kunden erstellten Konten mit dem Namen „Gast“.

Themen

- Aktivieren von Classic Queue v2 für Ihren RabbitMQ-Broker

Aktivieren Sie automatische Upgrades für kleinere Versionen

Verwenden Sie die neuesten Sicherheits- und Bugfixes sowie Leistungsverbesserungen der neuesten Broker-Version. Sie können automatische Upgrades für Nebenversionen für Amazon MQ aktivieren, um Upgrades auf die neueste Patch-Version zu verwalten.

Verwenden veralteter Funktionen

Wenn Sie Version 3.13 für RabbitMQ auf Amazon MQ verwenden, sehen Sie in der RabbitMQ Management UI ein Banner mit der Aufschrift: `Deprecated features are being used.`

Dies liegt daran, dass RabbitMQ auf Amazon MQ die folgenden Funktionen verwendet, die auf RabbitMQ nicht mehr angeboten werden oder automatisch für RabbitMQ auf Amazon MQ konfiguriert sind:

- Klassische Warteschlangenspiegelung
- Globales QoS
- Vorübergehende, nicht exklusive Warteschlangen

Dies ist ein Informationsbanner für Version 3.13, für das keine Aktion erforderlich ist. Ihr Amazon MQ-Broker wird diese Funktionen weiterhin verwenden.

Wählen Sie den richtigen Broker-Instance-Typ für den besten Durchsatz

Der Nachrichtendurchsatz eines Broker-Instance-Typs hängt von Ihrem Anwendungsfall ab. Kleinere Broker-Instance-Typen wie `t3.micro` sollten nur zum Testen der Anwendungsleistung verwendet werden. Wenn Sie diese Mikro-Instances verwenden, bevor Sie größere Instances in der Produktion einsetzen, können Sie die Anwendungsleistung verbessern und die Entwicklungskosten niedrig halten. Bei Instance-Typen `m5.large` und höher können Sie Cluster-Bereitstellungen für hohe Verfügbarkeit und Nachrichtenbeständigkeit verwenden. Größere Broker-Instance-Typen können das Produktionsniveau von Clients und Warteschlangen, einen hohen Durchsatz, Nachrichten im Speicher und redundante Nachrichten bewältigen. Weitere Informationen zur Auswahl des richtigen Instance-Typs finden Sie unter Richtlinien zur Größenbestimmung.

Verwenden Sie mehrere Kanäle

Verwenden Sie mehrere Kanäle über eine einzige Verbindung, um Verbindungsabwanderungen zu vermeiden. Anwendungen sollten ein Verhältnis von Verbindung zu Kanal von 1:1 vermeiden. Wir empfehlen, eine Verbindung pro Prozess und dann einen Kanal pro Thread zu verwenden. Vermeiden Sie eine übermäßige Kanalnutzung, um Kanallecks zu vermeiden.

Lazy-Warteschlangen aktivieren

Wenn Sie mit sehr langen Warteschlangen arbeiten, die große Mengen an Nachrichten verarbeiten, kann die Aktivierung verzögerter Warteschlangen die Leistung des Brokers verbessern.

Das Standardverhalten von RabbitMQ besteht darin, Nachrichten im Speicher zwischenzuspeichern und sie nur dann auf die Festplatte zu verschieben, wenn der Broker mehr verfügbaren Speicher benötigt. Das Verschieben von Nachrichten vom Speicher auf die Festplatte nimmt Zeit in Anspruch und stoppt die Nachrichtenverarbeitung. Lazy Queues beschleunigt den Prozess zwischen Speicher und Festplatte erheblich, da Nachrichten so schnell wie möglich auf der Festplatte gespeichert werden. Dadurch werden weniger Nachrichten im Arbeitsspeicher zwischengespeichert.

Sie können Lazy-Queues aktivieren, indem Sie die `queue.declare`-Argumente zum Zeitpunkt der Deklaration oder durch Konfigurieren einer Richtlinie über die RabbitMQ-Verwaltungskonsole. Das folgende Beispiel veranschaulicht das Deklarieren einer Lazy-Queue mit der RabbitMQ Java-Client-Bibliothek.

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

Alle Amazon MQ for RabbitMQ-Warteschlangen auf Version 3.12.13 und höher verhalten sich standardmäßig wie faule Warteschlangen. Informationen zum Upgrade auf die neueste Version von Amazon MQ für RabbitMQ finden Sie unter. ???

> **Note**
>
> Durch Aktivieren von Lazy Queues können Festplatten-I/O-Operationen erhöht werden.

Verwenden Sie persistente Nachrichten und dauerhafte Warteschlangen

Persistente Nachrichten können dazu beitragen, Datenverlust in Situationen zu verhindern, in denen ein Broker abstürzt oder neu gestartet wird. Persistente Nachrichten werden auf die Festplatte geschrieben, sobald sie eintreffen. Im Gegensatz zu Lazy Queues werden jedoch persistente Nachrichten sowohl im Arbeitsspeicher als auch auf der Festplatte zwischengespeichert, es sei denn, der Broker benötigt mehr Speicher. In Fällen, in denen mehr Speicher benötigt wird, werden Nachrichten vom RabbitMQ-Broker-Mechanismus aus dem Speicher entfernt, der das Speichern von Nachrichten auf der Festplatte verwaltet, allgemein alsSitzungspersistenz bezeichnet.

Um die Nachrichtenpersistenz zu aktivieren, können Sie Ihre Warteschlangen als`durable` erklären und den Nachrichtenübermittlungsmodus auf`persistent` stellen. Das folgende Beispiel veranschaulicht die Verwendung der RabbitMQ-Java-Client-Bibliothek, um eine dauerhafte Warteschlange zu deklarieren. Wenn Sie mit AMQP 0-9-1 arbeiten, können Sie Nachrichten als persistent markieren, indem Sie den Zustellungsmodus „2" einstellen.

```
boolean durable = true;
channel.queueDeclare("my_queue", durable, false, false, null);
```

Nachdem Sie die Warteschlange als dauerhaft konfiguriert haben, können Sie eine dauerhafte Nachricht an Ihre Warteschlange senden, indem Sie`MessageProperties` auf `PERSISTENT_TEXT_PLAIN` stellen, wie im folgenden Beispiel gezeigt.

```
import com.rabbitmq.client.MessageProperties;

channel.basicPublish("", "my_queue",
            MessageProperties.PERSISTENT_TEXT_PLAIN,
            message.getBytes());
```

Warteschlangen kurz halten

In Clusterbereitstellungen können Warteschlangen mit einer großen Anzahl von Nachrichten zu einer Überlastung der Ressourcen führen. Wenn ein Broker übermäßig ausgelastet ist, kann ein Neustart eines Amazon MQ für RabbitMQ Brokers zu weiteren Leistungseinbußen führen. Wenn ein Neustart durchgeführt wird, reagieren überlastete Broker möglicherweise nicht im `REBOOT_IN_PROGRESS` Zustand.

Während dem Wartungsfenster führt Amazon MQ alle Wartungsarbeiten jeweils einen Knoten aus, um sicherzustellen, dass der Broker betriebsbereit bleibt. Daher müssen Warteschlangen

möglicherweise synchronisiert werden, wenn jeder Knoten den Vorgang fortsetzt. Während der Synchronisation werden Nachrichten, die auf Spiegel repliziert werden müssen, vom entsprechenden Amazon Elastic Block Store (AmazonEBS) -Volume in den Speicher geladen und stapelweise verarbeitet. Durch die Verarbeitung von Nachrichten in Batches können Warteschlangen schneller synchronisiert werden.

Wenn Warteschlangen kurz gehalten werden und Nachrichten klein sind, werden die Warteschlangen erfolgreich synchronisiert und wie erwartet fortgesetzt. Wenn sich die Datenmenge in einem Batch jedoch dem Speicherlimit des Knotens nähert, löst der Knoten einen Alarm mit hohem Speicher aus, der die Warteschlangen-Synchronisierung pausiert. Sie können die Speichernutzung überprüfen, indem Sie die Metriken der Knoten `RabbitMemUsed` und des `RabbitMqMemLimit` Broker-Nodes unter vergleichen. CloudWatch Die Synchronisierung kann erst abgeschlossen werden, wenn Nachrichten verbraucht oder gelöscht oder die Anzahl der Nachrichten im Batch reduziert wird.

Wenn die Warteschlangensynchronisierung für eine Clusterbereitstellung angehalten wird, wird empfohlen, Nachrichten zu verwenden oder zu löschen, um die Anzahl der Nachrichten in Warteschlangen zu verringern. Sobald die Warteschlangentiefe reduziert und die Warteschlangensynchronisierung abgeschlossen ist, ändert sich der Broker-Status zu`RUNNING`. Um eine angehaltene Warteschlangensynchronisierung aufzulösen, können Sie eine Richtlinie auch aufReduzierung der Batch-Größe der Warteschlangensynchronisation anwenden.

Sie können auch automatische Löschvorgänge und TTL Richtlinien definieren, um den Ressourcenverbrauch proaktiv zu reduzieren und die Anzahl der Benutzer auf ein Minimum zu reduzieren. NACKs Das Warteschleifen von Nachrichten auf dem Broker ist CPU aufwändig, sodass eine hohe Anzahl von Nachrichten die Leistung des NACKs Brokers beeinträchtigen kann.

Bestätigung und Bestätigung konfigurieren

Wenn eine Client-Anwendung die Bestätigung der Zustellung und des Verbrauchs von Nachrichten an den Broker sendet, wird sie alsVerbraucherbestätigung bezeichnet. In ähnlicher Weise wird der Prozess der Bestätigung an einen Herausgeber alsVerlag bestätigen bezeichnet. Der Herausgeber bestätigt, dass Ihre Anwendung darüber informiert wird, wann Nachrichten zuverlässig gespeichert wurden. Ohne Bestätigung durch den Herausgeber akzeptiert Ihr Broker möglicherweise weiterhin Nachrichten, auch wenn der Speicherplatz knapp wird oder er sie nicht verarbeiten kann. Sowohl die Bestätigung als auch die Bestätigung sind unerlässlich, um die Datensicherheit bei der Arbeit mit RabbitMQ-Brokern zu gewährleisten.

Die Bestätigung der Verbraucherzustellung wird in der Regel in der Clientanwendung konfiguriert. Wenn Sie mit AMQP 0-9-1 arbeiten, kann die Bestätigung aktiviert werden, indem Sie das

`basic.consume` oder, wann eine Nachricht mit der Methode abgerufen wird, konfigurieren. `basic.code` AMQP0-9-1-Clients können auch Bestätigungen durch den Herausgeber konfigurieren, indem sie die Methode senden. `confirm.select`

In der Regel ist die Zustellungsbestätigung in einem Kanal aktiviert. Wenn Sie beispielsweise mit der RabbitMQ Java-Client-Bibliothek arbeiten, können Sie die`Channel#basicAck` verwenden, um eine einfache`basic.ack`Bestätigungsaufforderung erstellen, wie im folgenden Beispiel gezeigt.

```
// 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**
>
> Nicht bestätigte Nachrichten müssen im Speicher zwischengespeichert werden. Sie können die Anzahl der Nachrichten einschränken, die ein Konsumenten vorabruft, indem SieVorabruf-Einstellungen für eine Client-Anwendung konfigurieren.

Konfigurieren des Vorabrufs

Sie können den RabbitMQ-Prefetch-Wert verwenden, um zu optimieren, wie Ihre Verbraucher Nachrichten konsumieren. RabbitMQ implementiert den von AMQP 0-9-1 bereitgestellten Channel-Pre-Fetch-Mechanismus, indem der Pre-Fetch-Zähler auf Verbraucher und nicht auf Kanäle angewendet wird. Der Prefetch-Wert wird verwendet, um anzugeben, wie viele Nachrichten an den

Verbraucher zu einem bestimmten Zeitpunkt gesendet werden. Standardmäßig legt RabbitMQ eine unbegrenzte Puffergröße für Clientanwendungen fest.

Es gibt eine Vielzahl von Faktoren zu berücksichtigen, wenn Sie eine Pre-Fetch-Anzahl für Ihre RabbitMQ-Verbraucher festlegen. Berücksichtigen Sie zunächst die Umgebung und Konfiguration Ihrer Verbraucher. Da Verbraucher alle Nachrichten während der Verarbeitung im Speicher behalten müssen, kann ein hoher Pre-Fetch-Wert negative Auswirkungen auf die Leistung Ihrer Verbraucher haben und in einigen Fällen dazu führen, dass ein Verbraucher alle zusammen abstürzt. Ebenso behält der RabbitMQ-Broker selbst alle Nachrichten, die er im Speicher sendet, zwischengespeichert, bis er die Verbraucherbestätigung erhält. Ein hoher Prefetch-Wert kann dazu führen, dass Ihr RabbitMQ-Server schnell über den Arbeitsspeicher verfügt, wenn die automatische Bestätigung nicht für Verbraucher konfiguriert ist und wenn Verbraucher relativ lange Zeit benötigen, um Nachrichten zu verarbeiten.

In Anbetracht der obigen Überlegungen empfehlen wir, immer einen Pre-Fetch-Wert festzulegen, um Situationen zu vermeiden, in denen ein RabbitMQ-Broker oder seine Verbraucher aufgrund einer großen Anzahl von unverarbeiteten oder nicht bestätigten Nachrichten nicht genügend Arbeitsspeicher auslaufen. Wenn Sie Ihre Broker optimieren müssen, um große Mengen von Nachrichten zu verarbeiten, können Sie Ihre Broker und Verbraucher mit einer Reihe von Pre-Fetch-Zählungen testen, um den Wert zu bestimmen, an dem der Netzwerk-Overhead im Vergleich zu der Zeit, die ein Verbraucher benötigt, um Nachrichten zu verarbeiten, weitgehend unbedeutend wird.

> **Note**
>
> - Wenn Ihre Clientanwendungen so konfiguriert haben, dass die Zustellung von Nachrichten an Verbraucher automatisch bestätigt wird, hat das Festlegen eines Pre-Fetch-Werts keine Auswirkungen.
> - Alle vorab abgerufenen Nachrichten werden aus der Warteschlange entfernt.

Das folgende Beispiel demonstriert das Festlegen eines Vorabruf-Werts von`10`für einen einzelnen Verbraucher mit der RabbitMQ Java-Client-Bibliothek.

```
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**
>
> In der RabbitMQ-Java-Client-Bibliothek wird der Standardwert für die`global`-Flag auf`false` gestellt, so dass das obige Beispiel einfach als`channel.basicQos(10)` ausgeschrieben werden kann.

Konfigurieren von Celery

Python Celery sendet viele unnötige Nachrichten, die das Auffinden und Verarbeiten nützlicher Informationen erschweren können. Geben Sie den folgenden Befehl ein, um das Rauschen zu reduzieren und die Verarbeitung zu vereinfachen:

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

Automatische Wiederherstellung nach Netzwerkausfällen

Es wird empfohlen, die automatische Netzwerkwiederherstellung immer zu aktivieren, um erhebliche Ausfallzeiten zu vermeiden, wenn Clientverbindungen zu RabbitMQ-Knoten fehlschlagen. Die RabbitMQ Java-Client-Bibliothek unterstützt standardmäßig automatische Netzwerkwiederherstellung, beginnend mit Version`4.0.0`.

Die automatische Verbindungswiederherstellung wird ausgelöst, wenn eine nicht behandelte Ausnahme in der I/O-Schleife der Verbindung ausgelöst wird, wenn ein Timeout für den Socket-Lesevorgang erkannt wird oder wenn der Server eineHerzschlag verpasst.

In Fällen, in denen die anfängliche Verbindung zwischen einem Client und einem RabbitMQ-Knoten fehlschlägt, wird die automatische Wiederherstellung nicht ausgelöst. Wir empfehlen, Ihren Anwendungscode zu schreiben, um anfängliche Verbindungsfehler zu berücksichtigen, indem Sie die Verbindung erneut versuchen. Das folgende Beispiel veranschaulicht den erneuten Versuch von anfänglichen Netzwerkfehlern mithilfe der RabbitMQ-Java-Client-Bibliothek.

```
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**
>
> Wenn eine Anwendung eine Verbindung mit der`Connection.Close`-Methode wird die automatische Netzwerkwiederherstellung nicht aktiviert oder ausgelöst.

Aktivieren von Classic Queue v2 für Ihren RabbitMQ-Broker

Wir empfehlen, Classic Queue v2 (CQv2) auf den Broker-Engine-Versionen 3.10 und 3.11 zu aktivieren, um unter anderem folgende Leistungsverbesserungen zu erzielen:

- Verringern Sie den Speicherverbrauch
- Verbesserung der Verbraucherzustellung
- Erhöhung des Durchsatzes für Workloads, bei denen Verbraucher mit Produzenten Schritt halten

Alle Amazon MQ for RabbitMQ-Warteschlangen ab 3.12.13 werden standardmäßig verwendet. CQv2 Informationen zum Upgrade auf die neueste Version von Amazon MQ für RabbitMQ finden Sie unter. ???

Migration von zu CQv1 CQv2

Um es verwenden zu könnenCQv2, müssen Sie zuerst das `classic_mirrored_queue_version` Feature-Flag aktivieren. Weitere Informationen zu Feature-Flags finden Sie unter So aktivieren Sie Feature-Flags.

Um von CQv1 zu migrierenCQv2, müssen Sie eine neue Warteschlangenrichtlinie erstellen oder eine bestehende Warteschlangenrichtlinie bearbeiten, wobei die `queue-version`

Richtlinienschlüsseldefinition auf eingestellt ist2. Weitere Informationen zur Anwendung von Richtlinien finden Sie unterAnwenden von Richtlinien auf Amazon MQ für RabbitMQ. Weitere Informationen zur Aktivierung CQv2 mit einer Warteschlangenrichtlinie finden Sie unter Classic Queues in der RabbitMQ-Dokumentation.

Wir empfehlen, vor Beginn der Migration unsere anderen bewährten Methoden zur Leistungsoptimierung zu befolgen.

Wenn du eine Warteschlangenrichtlinie verwendest, führt das Löschen der Warteschlangenrichtlinie dazu, dass die Warteschlangen wieder heruntergestuft CQv2 werden. CQv1 Wir empfehlen nicht, Warteschlangen auf herunterzustufen, CQv1 da RabbitMQ die Darstellung der CQv2 Warteschlange auf der Festplatte konvertiert. Dies kann bei Warteschlangen mit großer Tiefe speicherintensiv und zeitaufwändig sein.

Sicherheit in Amazon MQ

Cloud-Sicherheit hat bei AWS höchste Priorität. Als AWS-Kunde profitieren Sie von Rechenzentren und Netzwerkarchitekturen, die eingerichtet wurden, um die Anforderungen der anspruchsvollsten Organisationen in puncto Sicherheit zu erfüllen.

Sicherheit ist eine übergreifende Verantwortlichkeit zwischen AWS und Ihnen. Das Modell der übergreifenden Verantwortlichkeit beschreibt dies als Sicherheit der Cloud und Sicherheit in der Cloud:

- Sicherheit der Cloud selbst – AWS ist dafür verantwortlich, die Infrastruktur zu schützen, mit der AWS-Services in der AWS Cloud ausgeführt werden. AWS stellt Ihnen außerdem Services bereit, die Sie sicher nutzen können. Auditoren von Drittanbietern testen und überprüfen die Effektivität unserer Sicherheitsmaßnahmen im Rahmen der AWS-Compliance-Programme regelmäßig. Informationen zu den Compliance-Programmen, die für Amazon MQ gelten, finden Sie unter Im Rahmen des Compliance-Programms zugelassene AWS-Services.
- Sicherheit in der Cloud – Ihr Verantwortungsumfang wird durch den AWS-Service bestimmt, den Sie verwenden. Sie sind auch für andere Faktoren verantwortlich, etwa für die Vertraulichkeit Ihrer Daten, für die Anforderungen Ihres Unternehmens und für die geltenden Gesetze und Vorschriften.

Diese Dokumentation zeigt Ihnen, wie Sie das Modell der übergreifenden Verantwortlichkeit bei der Verwendung von Amazon MQ einsetzen können. Die folgenden Themen veranschaulichen, wie Sie Amazon MQ zur Erfüllung Ihrer Sicherheits- und Compliance-Ziele konfigurieren können. Sie lernen auch, wie Sie andere AWS-Services verwenden, die Sie bei der Überwachung und Sicherung Ihrer Amazon MQ-Ressourcen unterstützen.

Themen

Datenschutz in Amazon MQ

Das AWS Modell der gilt für den Datenschutz in Amazon MQ. Wie in diesem Modell beschrieben, AWS ist es verantwortlich für den Schutz der globalen Infrastruktur, auf der AWS Cloud alle Systeme laufen. Sie sind dafür verantwortlich, die Kontrolle über Ihre in dieser Infrastruktur gehosteten Inhalte zu behalten. Sie sind auch für die Sicherheitskonfiguration und die Verwaltungsaufgaben für die von Ihnen verwendeten AWS -Services verantwortlich. Weitere Informationen zum Datenschutz finden Sie im Abschnitt Datenschutz FAQ. Informationen zum Datenschutz in Europa finden Sie im AWS Shared Responsibility Model und im GDPR Blogbeitrag im AWS Security Blog.

Aus Datenschutzgründen empfehlen wir, dass Sie Ihre AWS-Konto Anmeldeinformationen schützen und einzelne Benutzer mit AWS IAM Identity Center oder AWS Identity and Access Management (IAM) einrichten. So erhält jeder Benutzer nur die Berechtigungen, die zum Durchführen seiner Aufgaben erforderlich sind. Außerdem empfehlen wir, die Daten mit folgenden Methoden schützen:

- Verwenden Sie für jedes Konto eine Multi-Faktor-Authentifizierung (MFA).
- Verwenden SieSSL/TLS, um mit AWS Ressourcen zu kommunizieren. Wir benötigen TLS 1.2 und empfehlen TLS 1.3.
- Einrichtung API und Protokollierung von Benutzeraktivitäten mit AWS CloudTrail.
- Verwenden Sie AWS Verschlüsselungslösungen zusammen mit allen darin enthaltenen Standardsicherheitskontrollen AWS -Services.
- Verwenden Sie erweiterte verwaltete Sicherheitsservices wie Amazon Macie, die dabei helfen, in Amazon S3 gespeicherte persönliche Daten zu erkennen und zu schützen.
- Wenn Sie FIPS 140-3 validierte kryptografische Module für den Zugriff AWS über eine Befehlszeilenschnittstelle oder eine benötigenAPI, verwenden Sie einen Endpunkt. FIPS Weitere Informationen zu den verfügbaren FIPS Endpunkten finden Sie unter Federal Information Processing Standard () 140-3. FIPS

Wir empfehlen dringend, in Freitextfeldern, z. B. im Feld Name, keine vertraulichen oder sensiblen Informationen wie die E-Mail-Adressen Ihrer Kunden einzugeben. Dies gilt auch, wenn Sie mit Amazon MQ oder einem anderen AWS -Services über die Konsole arbeiten, API AWS CLI, oder AWS SDKs. Alle Daten, die Sie in Tags oder Freitextfelder eingeben, die für Namen verwendet werden, können für Abrechnungs- oder Diagnoseprotokolle verwendet werden. Wenn Sie einem externen Server eine URL zur Verfügung stellen, empfehlen wir dringend, dass Sie keine Anmeldeinformationen angeben, URL um Ihre Anfrage an diesen Server zu validieren.

Verwenden Sie bei Brokern von Amazon MQ for ActiveMQ und Amazon MQ for RabbitMQ keine personenbezogenen Daten (PII) oder andere vertrauliche oder sensible Informationen für Broker-Namen oder Benutzernamen, wenn Sie Ressourcen über die Broker-Webkonsole oder Amazon MQ erstellen. API Broker-Namen und Benutzernamen sind für andere Dienste zugänglich, einschließlich Logs. AWS CloudWatch Broker-Benutzernamen sind nicht für private oder sensible Daten gedacht.

Verschlüsselung

Die in Amazon MQ gespeicherten Benutzerdaten werden im Ruhezustand verschlüsselt. Amazon MQ Encryption at Rest bietet erhöhte Sicherheit, indem Ihre Daten mit Verschlüsselungsschlüsseln verschlüsselt werden, die in der AWS Key Management Service () KMS gespeichert sind. Dieser Service reduziert den Betriebsaufwand für den Schutz sensibler Daten sowie die Komplexität. Mit der Verschlüsselung von Daten im Ruhezustand können Sie sicherheitsrelevante Anwendungen erstellen, die Verschlüsselungsvorschriften und gesetzliche Bestimmungen einhalten.

Alle Verbindungen zwischen Amazon MQ-Brokern verwenden Transport Layer Security (TLS), um die Verschlüsselung während der Übertragung zu gewährleisten.

Amazon MQ verschlüsselt Nachrichten im Ruhezustand und unterwegs mit Verschlüsselungsschlüsseln, die es sicher verwaltet und speichert. Weitere Informationen finden Sie im AWS Encryption SDK -Entwicklerhandbuch.

Verschlüsselung im Ruhezustand

Amazon MQ ist in AWS Key Management Service (KMS) integriert, um eine transparente serverseitige Verschlüsselung zu bieten. Amazon MQ verschlüsselt Ihre Daten im Ruhezustand stets.

Wenn Sie einen Amazon MQ for ActiveMQ Broker oder einen Amazon MQ for RabbitMQ Broker erstellen, können Sie den Broker angeben AWS KMS key , den Amazon MQ zur Verschlüsselung Ihrer Daten im Ruhezustand verwenden soll. Wenn Sie keinen KMS Schlüssel angeben, erstellt Amazon MQ einen AWS eigenen KMS Schlüssel für Sie und verwendet ihn in Ihrem Namen. Amazon MQ unterstützt derzeit symmetrische SchlüsselKMS. Weitere Informationen zu KMS Schlüsseln finden Sie unter. AWS KMS keys

Beim Erstellen eines Brokers können Sie durch Auswahl einer der folgenden Optionen konfigurieren, was Amazon MQ als Verschlüsselungsschlüssel verwendet.

- Amazon MQ-eigener KMS Schlüssel (Standard) — Der Schlüssel gehört Amazon MQ und wird von Amazon MQ verwaltet und befindet sich nicht in Ihrem Konto.

- AWS verwalteter KMS Schlüssel — Der AWS verwaltete KMS Schlüssel (`aws/mq`) ist ein KMS Schlüssel in Ihrem Konto, der in Ihrem Namen von Amazon MQ erstellt, verwaltet und verwendet wird.
- Bestehenden, vom Kunden verwalteten KMS Schlüssel auswählen — Vom Kunden verwaltete KMS Schlüssel werden von Ihnen in AWS Key Management Service (KMS) erstellt und verwaltet.

Important

- Das Widerrufen einer Berechtigung kann nicht rückgängig gemacht werden. Stattdessen empfehlen wir, den Broker zu löschen, wenn Sie Zugriffsrechte widerrufen müssen.
- Für Amazon MQ for ActiveMQ-Broker, die Amazon Elastic File System (EFS) zum Speichern von Nachrichtendaten verwenden, gilt Folgendes: Wenn Sie die Erteilung, die Amazon die EFS Erlaubnis erteilt, die KMS Schlüssel in Ihrem Konto zu verwenden, widerrufen, erfolgt dies nicht sofort.
- Bei Brokern von Amazon MQ für RabbitMQ und Amazon MQ für ActiveMQ, die Nachrichtendaten speichern, kann Amazon Amazon MQ Ihren Broker nicht verwalten, wenn Sie die EBS Genehmigung, die Amazon die Nutzung der KMS Schlüssel in Ihrem Konto gewährt, deaktivieren, deren Löschung planen oder widerrufen, und er kann in einen heruntergekommenen Zustand übergehen. EBS
- Wenn Sie den Schlüssel deaktiviert oder das Löschen des Schlüssels geplant haben, können Sie den Schlüssel erneut aktivieren oder das Löschen des Schlüssels abbrechen und Ihren Broker weiter verwalten.
- Das Deaktivieren eines Schlüssels oder das Widerrufen einer Berechtigung erfolgt nicht sofort.

Wenn Sie einen Single-Instance-Broker mit einem KMS Schlüssel für RabbitMQ erstellen, werden Sie sehen, dass zwei Ereignisse angemeldet sind. `CreateGrant` AWS CloudTrail Bei der ersten Veranstaltung gewährt Amazon MQ einen Zuschuss für den KMS Schlüssel. Bei der zweiten Veranstaltung wird EBS ein Zuschuss EBS zur Verwendung bereitgestellt.

CreateGrant AWS CloudTrail Protokolleintrag: Single Instance Broker

mq_grant

```
{
    "eventVersion": "1.08",
    "userIdentity": {
        "type": "AssumedRole",
        "principalId": "AKIAIOSFODNN7EXAMPLE",
        "arn": "arn:aws:iam::111122223333:user/AmazonMqConsole",
        "accountId": "111122223333",
        "accessKeyId": "AKIAI44QH8DHBEXAMPLE",
        "sessionContext": {
            "sessionIssuer": {
                "type": "Role",
                "principalId": "AKIAIOSFODNN7EXAMPLE",
                "arn": "arn:aws:iam::111122223333:user/AmazonMqConsole",
                "accountId": "111122223333",
                "userName": "AmazonMqConsole"
            },
            "webIdFederationData": {},
            "attributes": {
                "creationDate": "2023-02-23T18:59:10Z",
                "mfaAuthenticated": "false"
            }
        },
        "invokedBy": "mq.amazonaws.com"
    },
    "eventTime": "2018-06-28T22:23:46Z",
    "eventSource": "amazonmq.amazonaws.com",
    "eventName": "CreateGrant",
    "awsRegion": "us-west-2",
    "sourceIPAddress": "203.0.113.0",
    "userAgent": "PostmanRuntime/7.1.5",
    "requestParameters": {
        "granteePrincipal": "mq.amazonaws.com",
        "keyId": "arn:aws:kms:us-east-1:316438333700:key/bdbe42ae-f825-4e78-
a8a1-828d411c4be2",
        "retiringPrincipal": "mq.amazonaws.com",
        "operations": [
            "CreateGrant",
            "Decrypt",
            "GenerateDataKeyWithoutPlaintext",
            "ReEncryptFrom",
            "ReEncryptTo",
            "DescribeKey"
        ]
    },
```

```
    "responseElements": {
        "grantId":
 "0ab0ac0d0b000f00ea00cc0a0e00fc00bce000c000f0000000c0bc0a0000aaafSAMPLE",
        "keyId": "arn:aws:kms:us-
west-2:111122223333:key/1234abcd-12ab-34cd-56ef-123456SAMPLE",

    "requestID": "ff000af-00eb-00ce-0e00-ea000fb0fba0SAMPLE",
    "eventID": "ff000af-00eb-00ce-0e00-ea000fb0fba0SAMPLE",
    "readOnly": false,
    "resources": [
        {
           "accountId": "111122223333",
            "type": "AWS::KMS::Key",
            "ARN": "arn:aws:kms:us-
west-2:111122223333:key/1234abcd-12ab-34cd-56ef-123456SAMPLE"
        }
    ],
    "eventType": "AwsApiCall",
    "managementEvent": true,
    "recipientAccountId": "111122223333",
    "eventCategory": "Management",
    "sessionCredentialFromConsole": "true"
}
```

EBS grant creation

Es wird ein Ereignis für die EBS Gewährung von Zuschüssen angezeigt.

```
                                {
    "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"
}
```


Wenn Sie eine Clusterbereitstellung mit einem KMS Schlüssel für RabbitMQ erstellen, werden fünf `CreateGrant` Ereignisse angemeldet angezeigt. AWS CloudTrail Bei den ersten beiden Ereignissen handelt es sich um das Erstellen von Erteilungen für Amazon MQ. Bei den nächsten drei Ereignissen handelt es sich um Zuschüsse, die von EBS for EBS to use erstellt wurden.

CreateGrant AWS CloudTrail Protokolleintrag: Cluster-Bereitstellung

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

Es werden drei Ereignisse für die Erstellung von EBS Zuschüssen angezeigt.

```
                                    {
    "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"
}
```


Weitere Informationen zu KMS Schlüsseln finden Sie AWS KMS keysim AWS Key Management Service Entwicklerhandbuch.

Verschlüsselung während der Übertragung

Amazon MQ for ActiveMQ: Amazon MQ for ActiveMQ erfordert eine starke Transport Layer Security (TLS) und verschlüsselt Daten, die zwischen den Brokern Ihrer Amazon MQ-Bereitstellung übertragen werden. Alle Daten, die zwischen Amazon MQ-Brokern übertragen werden, werden mit starker Transport Layer Security (TLS) verschlüsselt. Dies gilt für alle verfügbaren Protokolle.

Amazon MQ für RabbitMQ: Amazon MQ for RabbitMQ erfordert eine starke Transport Layer Security () -Verschlüsselung für alle Client-Verbindungen. TLS Der RabbitMQ-Cluster-Replikationsverkehr wird nur über den Ihres Brokers übertragen, VPC und der gesamte Netzwerkverkehr zwischen Rechenzentren wird auf der physischen Ebene transparent verschlüsselt. AWS Die geclusterten Broker von Amazon MQ für RabbitMQ unterstützen derzeit keine knotenübergreifende Verschlüsselung für die Cluster-Replikation. Weitere Informationen dazu finden Sie unter Verschlüsselung ruhender data-in-transit und übertragener Daten.

Amazon MQ für ActiveMQ Protokolle

Sie können mit den folgenden Protokollen auf Ihre ActiveMQ-Broker zugreifen, wenn diese aktiviert sind: TLS

- AMQP
- MQTT
- MQTTüber WebSocket
- OpenWire
- STOMP
- STOMPüber WebSocket

Unterstützte TLS Cipher Suites für ActiveMQ

ActiveMQ auf Amazon MQ unterstützt die folgenden Verschlüsselungs-Suiten:

- TLS_ _ _ _ ECDHE _256_ _ RSA WITH AES GCM SHA384
- TLS_ _ _ ECDHE _ RSA _256_ WITH _ AES CBC SHA384
- TLS_ _ _ ECDHE _ RSA _256_ WITH _ AES CBC SHA
- TLS_ _ _ DHE _ RSA _256_ WITH _ AES GCM SHA384
- TLS_ _ _ DHE _ RSA _256_ WITH _ AES CBC SHA256
- TLS_ _ _ DHE _ RSA _256_ WITH _ AES CBC SHA
- TLS_ _ _ RSA _256_ WITH _ AES GCM SHA384
- TLS_ _ _ RSA _256_ WITH _ AES CBC SHA256
- TLS_ _ _ RSA _256_ WITH _ AES CBC SHA
- TLS_ _ ECDHE _ _ RSA 128_ WITH _ AES GCM SHA256
- TLS_ _ ECDHE _ _ RSA 128_ WITH _ AES CBC SHA256
- TLS_ _ ECDHE _ _ RSA 128_ WITH _ AES CBC SHA
- TLS_ _ DHE _ _ RSA 128_ WITH _ AES GCM SHA256
- TLS_ _ DHE _ _ RSA 128_ WITH _ AES CBC SHA256
- TLS_ _ DHE _ _ RSA 128_ WITH _ AES CBC SHA
- TLS_ _ _ RSA _128_ WITH _ AES GCM SHA256

- TLS_ _ _ RSA _128_ WITH _ AES CBC SHA256
- TLS_ _ _ RSA _128_ WITH _ AES CBC SHA

Amazon MQ für RabbitMQ-Protokolle

Sie können mit den folgenden Protokollen auf Ihre RabbitMQ-Broker zugreifen, wenn diese aktiviert sind: TLS

- AMQP(0-9-1)

Unterstützte TLS Cipher Suites für RabbitMQ

RabbitMQ auf Amazon MQ unterstützt die folgenden Verschlüsselungs-Suiten:

- TLS_ _ _ _ _256_ _ ECDHE RSA WITH AES GCM SHA384
- TLS_ _ _ ECDHE _ RSA _128_ WITH _ AES GCM SHA256

Identitäts- und Zugriffsverwaltung für Amazon MQ

AWS Identity and Access Management (IAM) hilft einem Administrator AWS -Service , den Zugriff auf AWS Ressourcen sicher zu kontrollieren. IAMAdministratoren kontrollieren, wer authentifiziert (angemeldet) und autorisiert werden kann (über Berechtigungen verfügt), um Amazon MQ MQ-Ressourcen zu verwenden. IAMist eine AWS -Service , die Sie ohne zusätzliche Kosten verwenden können.

Themen

Zielgruppe

Wie Sie AWS Identity and Access Management (IAM) verwenden, hängt von der Arbeit ab, die Sie in Amazon MQ ausführen.

Service-Benutzer – Wenn Sie den Amazon MQ-Service zur Ausführung von Aufgaben verwenden, stellt Ihnen Ihr Administrator die Anmeldeinformationen und Berechtigungen bereit, die Sie benötigen. Wenn Sie zur Ausführung von Aufgaben weitere Amazon MQ-Funktionen verwenden, benötigen Sie möglicherweise zusätzliche Berechtigungen. Wenn Sie die Fuktionsweise der Zugriffskontrolle nachvollziehen, wissen Sie bereits, welche Berechtigungen Sie von Ihrem Administrator anzufordern müssen. Wenn Sie auf eine Funktion in Amazon MQ nicht zugreifen können, siehe Fehlerbehebung für Amazon MQ-Identität und -Zugriff.

Service-Administrator – Wenn Sie in Ihrem Unternehmen für die Amazon MQ-Ressourcen zuständig sind, haben Sie wahrscheinlich vollen Zugriff auf Amazon MQ. Ihre Aufgabe besteht darin, zu bestimmen, auf welche Amazon–MQ-Funktionen und -Ressourcen Ihre Service-Nutzer zugreifen sollen. Anschließend müssen Sie Anfragen an Ihren IAM Administrator senden, um die Berechtigungen Ihrer Servicebenutzer zu ändern. Lesen Sie die Informationen auf dieser Seite, um die grundlegenden Konzepte von zu verstehenIAM. Weitere Informationen darüber, wie Ihr Unternehmen Amazon MQ nutzen IAM kann, finden Sie unterSo funktioniert Amazon MQ mit IAM.

IAMAdministrator — Wenn Sie ein IAM Administrator sind, möchten Sie vielleicht mehr darüber erfahren, wie Sie Richtlinien zur Verwaltung des Zugriffs auf Amazon MQ schreiben können. Beispiele für identitätsbasierte Amazon MQ MQ-Richtlinien, die Sie in IAM verwenden können, finden Sie unter. Beispiele für identitätsbasierte Amazon MQ-Richtlinien

Authentifizierung mit Identitäten

Authentifizierung ist die Art und Weise, wie Sie sich AWS mit Ihren Identitätsdaten anmelden. Sie müssen sich als IAM Benutzer authentifizieren (angemeldet bei AWS) oder indem Sie eine IAM Rolle übernehmen. Root-Benutzer des AWS-Kontos

Sie können sich AWS als föderierte Identität anmelden, indem Sie Anmeldeinformationen verwenden, die über eine Identitätsquelle bereitgestellt wurden. AWS IAM Identity Center (IAMIdentity Center-) Nutzer, die Single-Sign-On-Authentifizierung Ihres Unternehmens und Ihre Google- oder Facebook-Anmeldeinformationen sind Beispiele für föderierte Identitäten. Wenn Sie sich als föderierte Identität anmelden, hat Ihr Administrator zuvor einen Identitätsverbund mithilfe von Rollen eingerichtet. IAM Wenn Sie AWS mithilfe eines Verbunds darauf zugreifen, übernehmen Sie indirekt eine Rolle.

Je nachdem, welcher Benutzertyp Sie sind, können Sie sich beim AWS Management Console oder beim AWS Zugangsportal anmelden. Weitere Informationen zur Anmeldung finden Sie AWS unter So melden Sie sich bei Ihrem an AWS-Konto im AWS-Anmeldung Benutzerhandbuch.

Wenn Sie AWS programmgesteuert darauf zugreifen, AWS stellt es ein Software Development Kit (SDK) und eine Befehlszeilenschnittstelle (CLI) bereit, mit der Sie Ihre Anfragen mithilfe Ihrer Anmeldeinformationen kryptografisch signieren können. Wenn Sie keine AWS Tools verwenden, müssen Sie Anfragen selbst signieren. Weitere Informationen zur Verwendung der empfohlenen Methode, um Anfragen selbst zu signieren, finden Sie im IAMBenutzerhandbuch unter AWS API Anfragen signieren.

Unabhängig von der verwendeten Authentifizierungsmethode müssen Sie möglicherweise zusätzliche Sicherheitsinformationen angeben. AWS Empfiehlt beispielsweise, die Multi-Faktor-Authentifizierung (MFA) zu verwenden, um die Sicherheit Ihres Kontos zu erhöhen. Weitere Informationen finden Sie unter Multi-Faktor-Authentifizierung im AWS IAM Identity Center Benutzerhandbuch und Verwenden der Multi-Faktor-Authentifizierung (MFA) AWS im IAM Benutzerhandbuch.

AWS-Konto Root-Benutzer

Wenn Sie einen erstellen AWS-Konto, beginnen Sie mit einer Anmeldeidentität, die vollständigen Zugriff auf alle AWS -Services Ressourcen im Konto hat. Diese Identität wird als AWS-Konto Root-Benutzer bezeichnet. Sie können darauf zugreifen, indem Sie sich mit der E-Mail-Adresse und dem Passwort anmelden, mit denen Sie das Konto erstellt haben. Wir raten ausdrücklich davon ab, den Root-Benutzer für Alltagsaufgaben zu verwenden. Schützen Sie Ihre Root-Benutzer-Anmeldeinformationen und verwenden Sie diese, um die Aufgaben auszuführen, die nur der Root-Benutzer ausführen kann. Eine vollständige Liste der Aufgaben, für die Sie sich als Root-Benutzer anmelden müssen, finden Sie im Benutzerhandbuch unter Aufgaben, für die Root-Benutzeranmeldedaten erforderlich sind. IAM

Benutzer und Gruppen

Ein IAMBenutzer ist eine Identität innerhalb Ihres Unternehmens AWS-Konto , die über spezifische Berechtigungen für eine einzelne Person oder Anwendung verfügt. Wir empfehlen, sich nach Möglichkeit auf temporäre Anmeldeinformationen zu verlassen, anstatt IAM Benutzer mit langfristigen Anmeldeinformationen wie Passwörtern und Zugriffsschlüsseln zu erstellen. Wenn Sie jedoch spezielle Anwendungsfälle haben, für die langfristige Anmeldeinformationen von IAM Benutzern erforderlich sind, empfehlen wir, die Zugriffsschlüssel abwechselnd zu verwenden. Weitere

Informationen finden Sie im Benutzerhandbuch unter Regelmäßiges Rotieren von Zugriffsschlüsseln für Anwendungsfälle, für die IAM langfristige Anmeldeinformationen erforderlich sind.

Eine IAMGruppe ist eine Identität, die eine Sammlung von IAM Benutzern angibt. Sie können sich nicht als Gruppe anmelden. Mithilfe von Gruppen können Sie Berechtigungen für mehrere Benutzer gleichzeitig angeben. Gruppen vereinfachen die Verwaltung von Berechtigungen, wenn es zahlreiche Benutzer gibt. Sie könnten beispielsweise eine Gruppe benennen IAMAdminsund dieser Gruppe Berechtigungen zur Verwaltung von IAM Ressourcen erteilen.

Benutzer unterscheiden sich von Rollen. Ein Benutzer ist einer einzigen Person oder Anwendung eindeutig zugeordnet. Eine Rolle kann von allen Personen angenommen werden, die sie benötigen. Benutzer besitzen dauerhafte Anmeldeinformationen. Rollen stellen temporäre Anmeldeinformationen bereit. Weitere Informationen finden Sie unter Wann sollte ein IAM Benutzer (statt einer Rolle) erstellt werden? im IAMBenutzerhandbuch.

IAMRollen

Eine IAMRolle ist eine Identität innerhalb von Ihnen AWS-Konto , für die bestimmte Berechtigungen gelten. Sie ähnelt einem IAM Benutzer, ist jedoch keiner bestimmten Person zugeordnet. Sie können vorübergehend eine IAM Rolle in der übernehmen, AWS Management Console indem Sie die Rollen wechseln. Sie können eine Rolle übernehmen, indem Sie eine AWS CLI AWS API OR-Operation aufrufen oder eine benutzerdefinierte Operation verwendenURL. Weitere Informationen zu Methoden zur Verwendung von Rollen finden Sie unter Verwenden von IAM Rollen im IAMBenutzerhandbuch.

IAMRollen mit temporären Anmeldeinformationen sind in den folgenden Situationen nützlich:

- Verbundbenutzerzugriff – Um einer Verbundidentität Berechtigungen zuzuweisen, erstellen Sie eine Rolle und definieren Berechtigungen für die Rolle. Wird eine Verbundidentität authentifiziert, so wird die Identität der Rolle zugeordnet und erhält die von der Rolle definierten Berechtigungen. Informationen zu Rollen für den Verbund finden Sie im IAMBenutzerhandbuch unter Erstellen einer Rolle für einen externen Identitätsanbieter. Wenn Sie IAM Identity Center verwenden, konfigurieren Sie einen Berechtigungssatz. Um zu kontrollieren, worauf Ihre Identitäten nach der Authentifizierung zugreifen können, korreliert IAM Identity Center den Berechtigungssatz mit einer Rolle in. IAM Informationen zu Berechtigungssätzen finden Sie unter Berechtigungssätze im AWS IAM Identity Center -Benutzerhandbuch.
- Temporäre IAM Benutzerberechtigungen — Ein IAM Benutzer oder eine Rolle kann eine IAM Rolle übernehmen, um vorübergehend verschiedene Berechtigungen für eine bestimmte Aufgabe zu übernehmen.

- Kontoübergreifender Zugriff — Sie können eine IAM Rolle verwenden, um jemandem (einem vertrauenswürdigen Principal) in einem anderen Konto den Zugriff auf Ressourcen in Ihrem Konto zu ermöglichen. Rollen stellen die primäre Möglichkeit dar, um kontoübergreifendem Zugriff zu gewähren. Bei einigen können Sie AWS -Services jedoch eine Richtlinie direkt an eine Ressource anhängen (anstatt eine Rolle als Proxy zu verwenden). Informationen zum Unterschied zwischen Rollen und ressourcenbasierten Richtlinien für den kontenübergreifenden Zugriff finden Sie IAMim Benutzerhandbuch unter Kontoübergreifender Ressourcenzugriff. IAM
- Serviceübergreifender Zugriff — Einige AWS -Services verwenden Funktionen in anderen. AWS -Services Wenn Sie beispielsweise in einem Service einen Anruf tätigen, ist es üblich, dass dieser Service Anwendungen in Amazon ausführt EC2 oder Objekte in Amazon S3 speichert. Ein Dienst kann dies mit den Berechtigungen des aufrufenden Prinzipals mit einer Servicerolle oder mit einer serviceverknüpften Rolle tun.
  - Zugriffssitzungen weiterleiten (FAS) — Wenn Sie einen IAM Benutzer oder eine Rolle verwenden, um Aktionen auszuführen AWS, gelten Sie als Principal. Bei einigen Services könnte es Aktionen geben, die dann eine andere Aktion in einem anderen Service initiieren. FASverwendet die Berechtigungen des Prinzipals, der an aufruft AWS -Service, kombiniert mit der Anforderung, Anfragen AWS -Service an nachgelagerte Dienste zu stellen. FASAnfragen werden nur gestellt, wenn ein Dienst eine Anfrage erhält, für deren Abschluss Interaktionen mit anderen AWS -Services oder Ressourcen erforderlich sind. In diesem Fall müssen Sie über Berechtigungen zum Ausführen beider Aktionen verfügen. Einzelheiten zu den Richtlinien beim Stellen von FAS Anfragen finden Sie unter Zugriffssitzungen weiterleiten.
  - Servicerolle — Eine Servicerolle ist eine IAMRolle, die ein Dienst übernimmt, um Aktionen in Ihrem Namen auszuführen. Ein IAM Administrator kann eine Servicerolle von innen heraus erstellen, ändern und löschenIAM. Weitere Informationen finden Sie im IAMBenutzerhandbuch unter Erstellen einer Rolle zum Delegieren von Berechtigungen AWS -Service an eine.
  - Dienstbezogene Rolle — Eine dienstverknüpfte Rolle ist eine Art von Servicerolle, die mit einer verknüpft ist. AWS -Service Der Service kann die Rolle übernehmen, um eine Aktion in Ihrem Namen auszuführen. Servicebezogene Rollen erscheinen in Ihrem Dienst AWS-Konto und gehören dem Dienst. Ein IAM Administrator kann die Berechtigungen für dienstbezogene Rollen anzeigen, aber nicht bearbeiten.
- Auf Amazon ausgeführte Anwendungen EC2 — Sie können eine IAM Rolle verwenden, um temporäre Anmeldeinformationen für Anwendungen zu verwalten, die auf einer EC2 Instance ausgeführt werden und AWS API Anfragen stellen AWS CLI . Dies ist dem Speichern von Zugriffsschlüsseln innerhalb der EC2 Instance vorzuziehen. Um einer EC2 Instanz eine AWS Rolle zuzuweisen und sie allen ihren Anwendungen zur Verfügung zu stellen, erstellen Sie ein

Instanzprofil, das an die Instanz angehängt ist. Ein Instanzprofil enthält die Rolle und ermöglicht Programmen, die auf der EC2 Instanz ausgeführt werden, temporäre Anmeldeinformationen abzurufen. Weitere Informationen finden Sie im IAMBenutzerhandbuch unter Verwenden einer IAM Rolle zur Erteilung von Berechtigungen für Anwendungen, die auf EC2 Amazon-Instances ausgeführt werden.

Informationen darüber, ob Sie IAM Rollen oder IAM Benutzer verwenden sollten, finden Sie im Benutzerhandbuch unter Wann sollte eine IAM Rolle (anstelle eines IAM Benutzers) erstellt werden.

Verwalten des Zugriffs mit Richtlinien

Sie steuern den Zugriff, AWS indem Sie Richtlinien erstellen und diese an AWS Identitäten oder Ressourcen anhängen. Eine Richtlinie ist ein Objekt, AWS das, wenn es einer Identität oder Ressource zugeordnet ist, deren Berechtigungen definiert. AWS wertet diese Richtlinien aus, wenn ein Prinzipal (Benutzer, Root-Benutzer oder Rollensitzung) eine Anfrage stellt. Berechtigungen in den Richtlinien bestimmen, ob die Anforderung zugelassen oder abgelehnt wird. Die meisten Richtlinien werden in AWS Form von JSON Dokumenten gespeichert. Weitere Informationen zur Struktur und zum Inhalt von JSON Richtliniendokumenten finden Sie im IAMBenutzerhandbuch unter Überblick über JSON Richtlinien.

Administratoren können mithilfe von AWS JSON Richtlinien festlegen, wer Zugriff auf was hat. Das bedeutet, welcher Prinzipal kann Aktionen für welche Ressourcen und unter welchen Bedingungen ausführen.

Standardmäßig haben Benutzer, Gruppen und Rollen keine Berechtigungen. Um Benutzern die Erlaubnis zu erteilen, Aktionen mit den Ressourcen durchzuführen, die sie benötigen, kann ein IAM Administrator IAM Richtlinien erstellen. Der Administrator kann dann die IAM Richtlinien zu Rollen hinzufügen, und Benutzer können die Rollen übernehmen.

IAMRichtlinien definieren Berechtigungen für eine Aktion, unabhängig von der Methode, mit der Sie den Vorgang ausführen. Angenommen, es gibt eine Richtlinie, die Berechtigungen für die `iam:GetRole`-Aktion erteilt. Ein Benutzer mit dieser Richtlinie kann Rolleninformationen aus dem AWS Management Console AWS CLI, dem oder dem abrufen AWS API.

Identitätsbasierte Richtlinien

Identitätsbasierte Richtlinien sind Dokumente mit JSON Berechtigungsrichtlinien, die Sie an eine Identität anhängen können, z. B. an einen IAM Benutzer, eine Benutzergruppe oder eine Rolle.

Diese Richtlinien steuern, welche Aktionen die Benutzer und Rollen für welche Ressourcen und unter welchen Bedingungen ausführen können. Informationen zum Erstellen einer identitätsbasierten Richtlinie finden Sie unter IAMRichtlinien erstellen im Benutzerhandbuch. IAM

Identitätsbasierte Richtlinien können weiter als Inline-Richtlinien oder verwaltete Richtlinien kategorisiert werden. Inline-Richtlinien sind direkt in einen einzelnen Benutzer, eine einzelne Gruppe oder eine einzelne Rolle eingebettet. Verwaltete Richtlinien sind eigenständige Richtlinien, die Sie mehreren Benutzern, Gruppen und Rollen in Ihrem System zuordnen können. AWS-Konto Zu den verwalteten Richtlinien gehören AWS verwaltete Richtlinien und vom Kunden verwaltete Richtlinien. Informationen dazu, wie Sie zwischen einer verwalteten Richtlinie oder einer Inline-Richtlinie wählen können, finden Sie im IAMBenutzerhandbuch unter Auswahl zwischen verwalteten Richtlinien und Inline-Richtlinien.

Ressourcenbasierte Richtlinien

Ressourcenbasierte Richtlinien sind JSON Richtliniendokumente, die Sie an eine Ressource anhängen. Beispiele für ressourcenbasierte Richtlinien sind IAM Rollenvertrauensrichtlinien und Amazon S3 S3-Bucket-Richtlinien. In Services, die ressourcenbasierte Richtlinien unterstützen, können Service-Administratoren sie verwenden, um den Zugriff auf eine bestimmte Ressource zu steuern. Für die Ressource, an welche die Richtlinie angehängt ist, legt die Richtlinie fest, welche Aktionen ein bestimmter Prinzipal unter welchen Bedingungen für diese Ressource ausführen kann. Sie müssen in einer ressourcenbasierten Richtlinie einen Prinzipal angeben. Zu den Prinzipalen können Konten, Benutzer, Rollen, Verbundbenutzer oder gehören. AWS -Services

Ressourcenbasierte Richtlinien sind Richtlinien innerhalb dieses Diensts. Sie können AWS verwaltete Richtlinien nicht IAM in einer ressourcenbasierten Richtlinie verwenden.

Zugriffskontrolllisten () ACLs

Zugriffskontrolllisten (ACLs) steuern, welche Principals (Kontomitglieder, Benutzer oder Rollen) über Zugriffsberechtigungen für eine Ressource verfügen. ACLsähneln ressourcenbasierten Richtlinien, verwenden jedoch nicht das JSON Richtliniendokumentformat.

Amazon S3 und AWS WAF Amazon VPC sind Beispiele für Dienste, die Unterstützung bietenACLs. Weitere Informationen finden Sie unter Übersicht über ACLs die Zugriffskontrollliste (ACL) im Amazon Simple Storage Service Developer Guide.

Weitere Richtlinientypen

AWS unterstützt zusätzliche, weniger verbreitete Richtlinientypen. Diese Richtlinientypen können die maximalen Berechtigungen festlegen, die Ihnen von den häufiger verwendeten Richtlinientypen erteilt werden können.

- Berechtigungsgrenzen — Eine Berechtigungsgrenze ist eine erweiterte Funktion, mit der Sie die maximalen Berechtigungen festlegen, die eine identitätsbasierte Richtlinie einer IAM Entität (IAMBenutzer oder Rolle) gewähren kann. Sie können eine Berechtigungsgrenze für eine Entität festlegen. Die daraus resultierenden Berechtigungen sind der Schnittpunkt der identitätsbasierten Richtlinien einer Entität und ihrer Berechtigungsgrenzen. Ressourcenbasierte Richtlinien, die den Benutzer oder die Rolle im Feld `Principal` angeben, werden nicht durch Berechtigungsgrenzen eingeschränkt. Eine explizite Zugriffsverweigerung in einer dieser Richtlinien setzt eine Zugriffserlaubnis außer Kraft. Weitere Informationen zu Berechtigungsgrenzen finden Sie im IAMBenutzerhandbuch unter Berechtigungsgrenzen für IAM Entitäten.
- Dienststeuerungsrichtlinien (SCPs) — SCPs sind JSON Richtlinien, die die maximalen Berechtigungen für eine Organisation oder Organisationseinheit (OU) in festlegen AWS Organizations. AWS Organizations ist ein Dienst zur Gruppierung und zentralen Verwaltung mehrerer AWS-Konten Unternehmenseigentümer. Wenn Sie alle Funktionen in einer Organisation aktivieren, können Sie Richtlinien zur Servicesteuerung (SCPs) auf einige oder alle Ihre Konten anwenden. Das SCP schränkt die Berechtigungen für Entitäten in Mitgliedskonten ein, einschließlich der einzelnen Entitäten Root-Benutzer des AWS-Kontos. Weitere Informationen zu Organizations und SCPs finden Sie unter Richtlinien zur Servicesteuerung im AWS Organizations Benutzerhandbuch.
- Sitzungsrichtlinien – Sitzungsrichtlinien sind erweiterte Richtlinien, die Sie als Parameter übergeben, wenn Sie eine temporäre Sitzung für eine Rolle oder einen verbundenen Benutzer programmgesteuert erstellen. Die resultierenden Sitzungsberechtigungen sind eine Schnittmenge der auf der Identität des Benutzers oder der Rolle basierenden Richtlinien und der Sitzungsrichtlinien. Berechtigungen können auch aus einer ressourcenbasierten Richtlinie stammen. Eine explizite Zugriffsverweigerung in einer dieser Richtlinien setzt eine Zugriffserlaubnis außer Kraft. Weitere Informationen finden Sie im IAMBenutzerhandbuch unter Sitzungsrichtlinien.

Mehrere Richtlinientypen

Wenn mehrere auf eine Anforderung mehrere Richtlinientypen angewendet werden können, sind die entsprechenden Berechtigungen komplizierter. Informationen darüber, wie AWS bestimmt

wird, ob eine Anfrage zulässig ist, wenn mehrere Richtlinientypen betroffen sind, finden Sie im IAMBenutzerhandbuch unter Bewertungslogik für Richtlinien.

So funktioniert Amazon MQ mit IAM

Bevor Sie IAM den Zugriff auf Amazon MQ verwalten, sollten Sie wissen, welche IAM Funktionen für Amazon MQ verfügbar sind. Einen umfassenden Überblick darüber, wie Amazon MQ und andere AWS Services zusammenarbeitenIAM, finden Sie unter AWS Services That Work with IAM im IAMBenutzerhandbuch.

Amazon MQ verwendet IAM zum Erstellen, Aktualisieren und Löschen von Vorgängen die native ActiveMQ-Authentifizierung für Makler. Weitere Informationen finden Sie unter Integration von ActiveMQ-Brokern in LDAP.

Themen

- Identitätsbasierte Amazon MQ-Richtlinien
- Ressourcenbasierte Amazon MQ -Richtlinien
- Autorisierung auf der Basis von Amazon MQ-Tags
- Amazon MQ-Rollen IAM

Identitätsbasierte Amazon MQ-Richtlinien

Mit IAM identitätsbasierten Richtlinien können Sie zulässige oder verweigerte Aktionen und Ressourcen sowie die Bedingungen angeben, unter denen Aktionen zugelassen oder verweigert werden. Amazon MQ unterstützt bestimmte Aktionen, Ressourcen und Bedingungsschlüssel. Weitere Informationen zu allen Elementen, die Sie in einer JSON Richtlinie verwenden, finden Sie im IAMBenutzerhandbuch unter IAMJSONPolicy Elements Reference.

Aktionen

Administratoren können mithilfe von AWS JSON Richtlinien angeben, wer Zugriff auf was hat. Das bedeutet, welcher Prinzipal kann Aktionen für welche Ressourcen und unter welchen Bedingungen ausführen.

Das `Action` Element einer JSON Richtlinie beschreibt die Aktionen, mit denen Sie den Zugriff in einer Richtlinie zulassen oder verweigern können. Richtlinienaktionen haben normalerweise denselben Namen wie der zugehörige AWS API Vorgang. Es gibt einige Ausnahmen, z. B. Aktionen, für die nur eine Genehmigung erforderlich ist und für die es keinen entsprechenden Vorgang gibt.

API Es gibt auch einige Operationen, die mehrere Aktionen in einer Richtlinie erfordern. Diese zusätzlichen Aktionen werden als abhängige Aktionen bezeichnet.

Schließen Sie Aktionen in eine Richtlinie ein, um Berechtigungen zur Durchführung der zugeordneten Operation zu erteilen.

Richtlinienaktionen in Amazon MQ verwenden das folgende Präfix vor der Aktion: `mq:`. Um beispielsweise jemandem die Erlaubnis zu erteilen, eine Amazon MQ-Instance mit dem Amazon MQ `CreateBroker` API MQ-Vorgang auszuführen, nehmen Sie die `mq:CreateBroker` Aktion in seine Richtlinie auf. Richtlinienanweisungen müssen entweder ein – `Action`oder ein `NotAction`-Element enthalten. Amazon MQ definiert eine eigene Gruppe von Aktionen, die Aufgaben beschreiben, die Sie mit diesem Service durchführen können.

Um mehrere Aktionen in einer einzigen Anweisung anzugeben, trennen Sie sie wie folgt durch Kommata:

```
"Action": [
      "mq:action1",
      "mq:action2"
```

Sie können auch Platzhalter verwenden, um mehrere Aktionen anzugeben. Beispielsweise können Sie alle Aktionen festlegen, die mit dem Wort `Describe` beginnen, einschließlich der folgenden Aktion:

```
"Action": "mq:Describe*"
```

Eine Liste der Amazon MQ-Aktionen finden Sie unter Von Amazon MQ definierte Aktionen im IAMBenutzerhandbuch.

Ressourcen

Administratoren können mithilfe von AWS JSON Richtlinien festlegen, wer Zugriff auf was hat. Das bedeutet, welcher Prinzipal kann Aktionen für welche Ressourcen und unter welchen Bedingungen ausführen.

Das `Resource` JSON Richtlinienelement gibt das Objekt oder die Objekte an, für die die Aktion gilt. Anweisungen müssen entweder ein – `Resource`oder ein `NotResource`-Element enthalten. Es hat sich bewährt, eine Ressource mit ihrem Amazon-Ressourcennamen (ARN) anzugeben. Sie können dies für Aktionen tun, die einen bestimmten Ressourcentyp unterstützen, der als Berechtigungen auf Ressourcenebene bezeichnet wird.

Verwenden Sie für Aktionen, die keine Berechtigungen auf Ressourcenebene unterstützen, z. B. Auflistungsoperationen, einen Platzhalter (*), um anzugeben, dass die Anweisung für alle Ressourcen gilt.

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

Im Amazon MQ sind die primären AWS Ressourcen ein Amazon MQ MQ-Nachrichtenbroker und dessen Konfiguration. Amazon MQ-Brokern und Konfigurationen sind jeweils eindeutige Amazon-Ressourcennamen (ARNs) zugeordnet, wie in der folgenden Tabelle dargestellt.

Ressource ntypen	ARN	Bedingungsschlüssel
brokers	`arn:aws:mq:us-east-1:123456789012:broker:${brokerName}:${brokerId}`	aws:ResourceTag/${TagKey}
configurations	`arn:${Partition}:mq:${Region}:${Account}:configuration:${configuration-id}`	aws:ResourceTag/${TagKey}

Weitere Informationen zum Format von ARNs finden Sie unter Amazon Resource Names (ARNs) und AWS Service Namespaces.

Um beispielsweise den Makler anzugeben, `MyBroker` mit dem brokerId `b-1234a5b6-78cd-901e-2fgh-3i45j6k178l9` in Ihrem Kontoauszug benannt ist, verwenden Sie Folgendes: ARN

```
"Resource": "arn:aws:mq:us-
east-1:123456789012:broker:MyBroker:b-1234a5b6-78cd-901e-2fgh-3i45j6k178l9"
```

Um alle Broker und Konfigurationen anzugeben, die zu einem bestimmten Konto gehören, verwenden Sie den Platzhalter (*):

```
"Resource": "arn:aws:mq:us-east-1:123456789012:*"
```

Einige Amazon MQ-Aktionen, z. B. das Erstellen von Ressourcen, können auf bestimmten Ressourcen nicht ausgeführt werden. In diesen Fällen müssen Sie den Platzhalter (*) verwenden.

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

Für die API Aktion `CreateTags` sind sowohl ein Broker als auch eine Konfiguration erforderlich. Um mehrere Ressourcen in einer einzigen Anweisung anzugeben, trennen Sie sie ARNs durch Kommas.

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

Eine Liste der Amazon MQ-Ressourcentypen und ihrer Eigenschaften ARNs finden Sie unter Von Amazon MQ definierte Ressourcen im IAMBenutzerhandbuch. Informationen darüber, mit welchen Aktionen Sie die ARN einzelnen Ressourcen spezifizieren können, finden Sie unter Von Amazon MQ definierte Aktionen.

Bedingungsschlüssel

Administratoren können mithilfe von AWS JSON Richtlinien festlegen, wer Zugriff auf was hat. Das heißt, welcher Prinzipal kann Aktionen für welche Ressourcen und unter welchen Bedingungen ausführen.

Das Element `Condition` (oder `Condition` block) ermöglicht Ihnen die Angabe der Bedingungen, unter denen eine Anweisung wirksam ist. Das Element `Condition` ist optional. Sie können bedingte Ausdrücke erstellen, die Bedingungsoperatoren verwenden, z. B. ist gleich oder kleiner als, damit die Bedingung in der Richtlinie mit Werten in der Anforderung übereinstimmt.

Wenn Sie mehrere `Condition`-Elemente in einer Anweisung oder mehrere Schlüssel in einem einzelnen `Condition`-Element angeben, wertet AWS diese mittels einer logischen `AND`-Operation aus. Wenn Sie mehrere Werte für einen einzelnen Bedingungsschlüssel angeben, AWS wertet die Bedingung mithilfe einer logischen `OR` Operation aus. Alle Bedingungen müssen erfüllt werden, bevor die Berechtigungen der Anweisung gewährt werden.

Sie können auch Platzhaltervariablen verwenden, wenn Sie Bedingungen angeben. Sie können einem IAM Benutzer beispielsweise nur dann Zugriff auf eine Ressource gewähren, wenn sie mit seinem IAM Benutzernamen gekennzeichnet ist. Weitere Informationen finden Sie im IAMBenutzerhandbuch unter IAMRichtlinienelemente: Variablen und Tags.

AWS unterstützt globale Bedingungsschlüssel und dienstspezifische Bedingungsschlüssel. Eine Übersicht aller AWS globalen Bedingungsschlüssel finden Sie unter Kontext-Schlüssel für AWS globale Bedingungen im IAMBenutzerhandbuch.

Amazon MQ stellt keine servicespezifischen Bedingungsschlüssel bereit, unterstützt jedoch die Verwendung einiger globaler Bedingungsschlüssel. Eine Liste der Amazon MQ-Bedingungsschlüssel finden Sie in der Tabelle unten oder in der IAMBedienungsanleitung unter Condition Keys for Amazon MQ. Informationen dazu, mit welchen Aktionen und Ressourcen Sie einen Bedingungsschlüssel verwenden können, finden Sie unter Von Amazon MQ definierte Aktionen.

Bedingungsschlüssel	Beschreibung	Typ
aws: RequestTag /$ {} TagKey	Filtert Aktionen basierend auf den Tags, die in der Anforderung übergeben werden.	String
as: ResourceTag /$ {} TagKey	Filtert Aktionen basierend auf den Tags, die der Ressource zugeordnet sind.	String
war: TagKeys	Filtert Aktionen basierend auf den Tag-Schlüsseln, die in der Anforderung übergeben werden.	String

Beispiele

Beispiele für identitätsbasierte Amazon MQ-Richtlinien finden Sie unter .

Ressourcenbasierte Amazon MQ -Richtlinien

Derzeit unterstützt Amazon MQ keine IAM Authentifizierung mit ressourcenbasierten Berechtigungen oder ressourcenbasierten Richtlinien.

Autorisierung auf der Basis von Amazon MQ-Tags

Sie können Tags an Amazon MQ-Ressourcen anhängen oder Tags in einer Anforderung an Amazon MQ übergeben. Um den Zugriff auf der Grundlage von Tags zu steuern, geben Sie im Bedingungselement einer Richtlinie Informationen an, indem Sie die Bedingungsschlüssel `mq:ResourceTag/`*`key-name`*, `aws:RequestTag/`*`key-name`*, oder `aws:TagKeys` verwenden.

Amazon MQ unterstützt Richtlinien, die auf Tags basieren. Sie können z. B. den Zugriff auf alle Amazon MQ-Ressourcen einschränken, die ein Tag mit dem Schlüssel `environment` und dem Wert `production` enthalten:

```
{
    "Version": "2012-10-17",
    "Statement": [
        {
            "Effect": "Deny",
            "Action": [
                "mq:DeleteBroker",
                "mq:RebootBroker",
                "mq:DeleteTags"
            ],
            "Resource": "*",
            "Condition": {
                "StringEquals": {
                    "aws:ResourceTag/environment": "production"
                }
            }
        }
    ]
}
```

Mit dieser Richtlinie `Deny` Sie die Möglichkeit, einen Amazon-MQ-Broker zu löschen oder neu zu starten, der das Tag `environment/production` enthält.

Weitere Informationen zum Markieren finden Sie unter:

- Hinzufügen von Tags zu Amazon MQ MQ-Ressourcen
- IAMSteuern des Zugriffs mithilfe von Tags

Amazon MQ-Rollen IAM

Eine IAMRolle ist eine Entität innerhalb Ihres AWS Kontos, die über bestimmte Berechtigungen verfügt.

Verwenden temporärer Anmeldeinformationen mit Amazon MQ

Sie können temporäre Anmeldeinformationen verwenden, um sich bei Federation anzumelden, eine IAM Rolle zu übernehmen oder eine kontoübergreifende Rolle anzunehmen. Sie erhalten temporäre Sicherheitsanmeldedaten, indem Sie AWS STS API Operationen wie AssumeRoleoder GetFederationTokenaufrufen.

Amazon MQ unterstützt die Verwendung temporärer Anmeldeinformationen.

Servicerollen

Dieses Feature ermöglicht einem Service das Annehmen einer Servicerolle in Ihrem Namen. Diese Rolle gewährt dem Service Zugriff auf Ressourcen in anderen Diensten, um eine Aktion in Ihrem Namen auszuführen. Servicerollen werden in Ihrem IAM Konto angezeigt und gehören dem Konto. Das bedeutet, dass ein IAM Administrator die Berechtigungen für diese Rolle ändern kann. Dies kann jedoch die Funktionalität des Dienstes beeinträchtigen.

Amazon MQ unterstützt Servicerollen.

Beispiele für identitätsbasierte Amazon MQ-Richtlinien

Benutzer und Rollen besitzen standardmäßig keine Berechtigungen zum Erstellen oder Ändern von Amazon-MQ-Ressourcen. Sie können auch keine Aufgaben ausführen, die die AWS Management Console-, AWS CLI- oder AWS-API benutzen. Ein IAM-Administrator muss IAM-Richtlinien erstellen, die Benutzern und Rollen die Berechtigung zum Ausführen bestimmter API-Operationen für die angegebenen Ressourcen gewähren, die diese benötigen. Der Administrator muss diese Richtlinien anschließend den IAM-Benutzern oder -Gruppen anfügen, die diese Berechtigungen benötigen.

Informationen dazu, wie Sie unter Verwendung dieser beispielhaften JSON-Richtliniendokumente eine identitätsbasierte IAM-Richtlinie erstellen, finden Sie unter Erstellen von Richtlinien auf der JSON-Registerkarte im IAM-Benutzerhandbuch.

Themen

- Bewährte Methoden für Richtlinien
- Verwenden der Amazon MQ-Konsole
- Gewähren der Berechtigung zur Anzeige der eigenen Berechtigungen für Benutzer

Bewährte Methoden für Richtlinien

Identitätsbasierte Richtlinien können festlegen, ob jemand Amazon–MQ-Ressourcen in Ihrem Konto erstellen, darauf zugreifen oder daraus löschen kann. Dies kann zusätzliche Kosten für Ihr verursachen AWS-Konto. Befolgen Sie beim Erstellen oder Bearbeiten identitätsbasierter Richtlinien die folgenden Anleitungen und Empfehlungen:

- Erste Schritte mit AWS-verwaltete Richtlinien und Umstellung auf Berechtigungen mit den geringsten Berechtigungen – Um Ihren Benutzern und Workloads Berechtigungen zu gewähren, verwenden Sie die AWS-verwaltete Richtlinien die Berechtigungen für viele allgemeine

Anwendungsfälle gewähren. Sie sind in Ihrem AWS-Konto verfügbar. Wir empfehlen Ihnen, die Berechtigungen weiter zu reduzieren, indem Sie AWS-kundenverwaltete Richtlinien definieren, die speziell auf Ihre Anwendungsfälle zugeschnitten sind. Weitere Informationen finden Sie unter AWS-verwaltete Richtlinien oder AWS-verwaltete Richtlinien für Auftragsfunktionen im IAM-Benutzerhandbuch.

- Anwendung von Berechtigungen mit den geringsten Rechten – Wenn Sie mit IAM-Richtlinien Berechtigungen festlegen, gewähren Sie nur die Berechtigungen, die für die Durchführung einer Aufgabe erforderlich sind. Sie tun dies, indem Sie die Aktionen definieren, die für bestimmte Ressourcen unter bestimmten Bedingungen durchgeführt werden können, auch bekannt als die geringsten Berechtigungen. Weitere Informationen zur Verwendung von IAM zum Anwenden von Berechtigungen finden Sie unter Richtlinien und Berechtigungen in IAM im IAM-Benutzerhandbuch.
- Verwenden von Bedingungen in IAM-Richtlinien zur weiteren Einschränkung des Zugriffs – Sie können Ihren Richtlinien eine Bedingung hinzufügen, um den Zugriff auf Aktionen und Ressourcen zu beschränken. Sie können beispielsweise eine Richtlinienbedingung schreiben, um festzulegen, dass alle Anforderungen mithilfe von SSL gesendet werden müssen. Sie können auch Bedingungen verwenden, um Zugriff auf Service-Aktionen zu gewähren, wenn diese durch ein bestimmtes AWS -Service, wie beispielsweise AWS CloudFormation, verwendet werden. Weitere Informationen finden Sie unter IAM-JSON-Richtlinienelemente: Bedingung im IAM-Benutzerhandbuch.
- Verwenden von IAM Access Analyzer zur Validierung Ihrer IAM-Richtlinien, um sichere und funktionale Berechtigungen zu gewährleisten – IAM Access Analyzer validiert neue und vorhandene Richtlinien, damit die Richtlinien der IAM-Richtliniensprache (JSON) und den bewährten IAM-Methoden entsprechen. IAM Access Analyzer stellt mehr als 100 Richtlinienprüfungen und umsetzbare Empfehlungen zur Verfügung, damit Sie sichere und funktionale Richtlinien erstellen können. Weitere Informationen finden Sie unter Richtlinienvalidierung zum IAM Access Analyzer im IAM-Benutzerhandbuch.
- Bedarf einer Multi-Faktor-Authentifizierung (MFA) – Wenn Sie ein Szenario haben, das IAM-Benutzer oder Root-Benutzer in Ihrem AWS-Konto erfordert, aktivieren Sie MFA für zusätzliche Sicherheit. Um MFA beim Aufrufen von API-Vorgängen anzufordern, fügen Sie Ihren Richtlinien MFA-Bedingungen hinzu. Weitere Informationen finden Sie unter Konfigurieren eines MFA-geschützten API-Zugriffs im IAM-Benutzerhandbuch.

Weitere Informationen zu bewährten Methoden in IAM finden Sie unter Bewährte Methoden für die Sicherheit in IAM im IAM-Benutzerhandbuch.

Verwenden der Amazon MQ-Konsole

Um auf die Amazon MQ-Konsole zugreifen zu können, müssen Sie über einen Mindestsatz von Berechtigungen verfügen. Diese Berechtigungen müssen Ihnen das Auflisten und Anzeigen von Details zu Amazon MQ-Ressourcen in Ihrem AWS-Konto gestatten. Wenn Sie eine identitätsbasierte Richtlinie erstellen, die strenger ist als die mindestens erforderlichen Berechtigungen, funktioniert die Konsole nicht wie vorgesehen für Entitäten (IAM-Benutzer oder -Rollen) mit dieser Richtlinie.

Um sicherzustellen, dass diese Entitäten dennoch die Amazon MQ-Konsole verwenden können, fügen Sie den Entitäten auch die folgende von AWS verwaltete Richtlinie an. Weitere Informationen finden Sie unter Hinzufügen von Berechtigungen zu einem Benutzer im IAM-Benutzerhandbuch.

```
AmazonMQReadOnlyAccess
```

Für Benutzer, die nur Aufrufe an die AWS CLI oder AWS-API durchführen, müssen Sie keine Mindestberechtigungen in der Konsole erteilen. Stattdessen sollten Sie nur Zugriff auf die Aktionen zulassen, die den API-Operation entsprechen, die Sie ausführen möchten.

Gewähren der Berechtigung zur Anzeige der eigenen Berechtigungen für Benutzer

In diesem Beispiel wird gezeigt, wie Sie eine Richtlinie erstellen, die IAM-Benutzern die Berechtigung zum Anzeigen der eingebundenen Richtlinien und verwalteten Richtlinien gewährt, die ihrer Benutzeridentität angefügt sind. Diese Richtlinie enthält Berechtigungen für die Ausführung dieser Aktion auf der Konsole oder für die programmgesteuerte Ausführung über die AWS CLI oder die AWS-API.

```
{
    "Version": "2012-10-17",
    "Statement": [
        {
            "Sid": "ViewOwnUserInfo",
            "Effect": "Allow",
            "Action": [
                "iam:GetUserPolicy",
                "iam:ListGroupsForUser",
                "iam:ListAttachedUserPolicies",
                "iam:ListUserPolicies",
                "iam:GetUser"
            ],
            "Resource": ["arn:aws:iam::*:user/${aws:username}"]
```

```
        },
        {
            "Sid": "NavigateInConsole",
            "Effect": "Allow",
            "Action": [
                "iam:GetGroupPolicy",
                "iam:GetPolicyVersion",
                "iam:GetPolicy",
                "iam:ListAttachedGroupPolicies",
                "iam:ListGroupPolicies",
                "iam:ListPolicyVersions",
                "iam:ListPolicies",
                "iam:ListUsers"
            ],
            "Resource": "*"
        }
    ]
}
```

API-Authentifizierung und Amazon MQ-Autorisierung für

Amazon MQ verwendet die standardmäßige AWS-Anforderungssignatur für die API-Authentifizierung. Weitere Informationen dazu finden Sie unter Signieren von AWS API-Anforderungen im Allgemeine AWS-Referenz.

> **Note**
>
> Derzeit unterstützt Amazon MQ keine IAM-Authentifizierung unter Verwendung ressourcenbasierter Berechtigungen oder ressourcenbasierter Richtlinien.

Um AWS-Benutzer für die Arbeit mit Brokern, Konfigurationen und Benutzern zu autorisieren, müssen Sie die IAM-Richtlinienberechtigungen bearbeiten.

Themen

- Erforderliche IAM-Berechtigungen zum Erstellen eines Amazon MQ-Brokers
- Amazon MQ REST API-Berechtigungen–Referenz
- Unterstützte Berechtigungen auf Ressourcenebene für Amazon MQ-API-Aktionen

Erforderliche IAM-Berechtigungen zum Erstellen eines Amazon MQ-Brokers

Um einen Broker zu erstellen, müssen Sie entweder die `AmazonMQFullAccess`-IAM-Richtlinie verwenden oder die folgenden EC2-Berechtigungen in Ihre IAM-Richtlinie aufnehmen.

Die folgende benutzerdefinierte Richtlinie besteht aus zwei Anweisungen (eine bedingte), die Berechtigungen zum Ändern der Ressourcen erteilen, die Amazon MQ benötigt, um einen ActiveMQ-Broker zu erstellen.

Important

- Die `ec2:CreateNetworkInterface`-Aktion ist erforderlich, damit Amazon MQ eine Elastic Network-Schnittstelle (Elastic Network Interface, ENI) in Ihrem Konto für Sie erstellen kann.
- Die `ec2:CreateNetworkInterfacePermission`-Aktion erlaubt es Amazon MQ, die ENI an einen ActiveMQ-Broker anzufügen.
- Der `ec2:AuthorizedService`-Bedingungsschlüssel stellt sicher, dass ENI-Berechtigungen nur Amazon MQ-Service-Konten gewährt werden.

```
{
    "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"
            }
        }
    }]
}
```

Weitere Informationen finden Sie unter Schritt 2: Erstellen Sie einen Benutzer und holen Sie sich Ihre Anmeldeinformationen AWS und Verändern oder löschen Sie auf keinen Fall die Amazon MQ Elastic Network-Schnittstelle.

Amazon MQ REST API-Berechtigungen–Referenz

Die folgende Tabelle listet Amazon MQ REST-APIs und die entsprechenden IAM-Berechtigungen auf.

Amazon MQ REST-APIs und erforderliche Berechtigungen

Amazon MQ REST APIs	Erforderliche Berechtigungen
CreateBroker	mq:CreateBroker
CreateConfiguration	mq:CreateConfiguration
CreateTags	mq:CreateTags
CreateUser	mq:CreateUser
DeleteBroker	mq:DeleteBroker
DeleteUser	mq:DeleteUser
DescribeBroker	mq:DescribeBroker
DescribeConfiguration	mq:DescribeConfiguration

Amazon MQ REST APIs	Erforderliche Berechtigungen
DescribeConfigurationRevision	mq:DescribeConfigurationRevision
DescribeUser	mq:DescribeUser
ListBrokers	mq:ListBrokers
ListConfigurationRevisions	mq:ListConfigurationRevisions
ListConfigurations	mq:ListConfigurations
ListTags	mq:ListTags
ListUsers	mq:ListUsers
RebootBroker	mq:RebootBroker
UpdateBroker	mq:UpdateBroker
UpdateConfiguration	mq:UpdateConfiguration
UpdateUser	mq:UpdateUser

Unterstützte Berechtigungen auf Ressourcenebene für Amazon MQ-API-Aktionen

Berechtigungen auf Ressourcenebene bedeutet, dass Sie angeben können, für welche Ressourcen die Benutzer Aktionen ausführen dürfen. Amazon MQ unterstützt teilweise Berechtigungen auf Ressourcenebene. Bei bestimmten Amazon MQ-Aktionen können Sie kontrollieren, wann die Benutzer diese Aktionen verwenden dürfen. Dies basiert auf Bedingungen, die erfüllt sein müssen, oder auf bestimmten Ressourcen, die von den Benutzern verwendet werden dürfen.

In der folgenden Tabelle werden die Amazon MQ-API-Aktionen aufgeführt, die Berechtigungen auf Ressourcenebene derzeit unterstützen, sowie die unterstützten Ressourcen, Ressourcen-ARNs und Bedingungsschlüssel für jede Aktion.

> **Important**
>
> Falls eine Amazon MQ-API-Aktion nicht in dieser Tabelle genannt wird, unterstützt sie keine Berechtigungen auf Ressourcenebene. Wenn eine Amazon MQ-API-Aktion Berechtigungen

auf Ressourcenebene nicht unterstützt, können Sie den Benutzern die Berechtigung zur Verwendung dieser Aktion erteilen, müssen aber für das Ressourcenelement in der Richtlinienanweisung ein Sternchen * als Platzhalterzeichen einfügen.

API-Aktion	Ressourcentypen (*erforderlich)
CreateConfiguration	Konfigurationen*
CreateTags	Broker, Konfigurationen
CreateUser	Broker*
DeleteBroker	Broker*
DeleteUser	Broker*
DescribeBroker	Broker*
DescribeConfiguration	Konfigurationen*
DescribeConfigurationRevision	Konfigurationen*
DescribeUser	Broker*
ListConfigurationRevisions	Konfigurationen*
ListConfigurationRevisions	Konfigurationen*
ListTags	Broker, Konfigurationen
ListUsers	Broker*
RebootBroker	Broker*
UpdateBroker	Broker*

API-Aktion	Ressourcentypen (*erforderlich)
UpdateConfiguration	Konfigurationen*
UpdateUser	Broker*

AWS verwaltete Richtlinien für Amazon MQ

Eine AWS verwaltete Richtlinie ist eine eigenständige Richtlinie, die von erstellt und verwaltet AWS wird. AWS Verwaltete Richtlinien sind so konzipiert, dass sie Berechtigungen für viele gängige Anwendungsfälle bereitstellen, sodass Sie damit beginnen können, Benutzern, Gruppen und Rollen Berechtigungen zuzuweisen.

Beachten Sie, dass AWS verwaltete Richtlinien für Ihre speziellen Anwendungsfälle möglicherweise keine Berechtigungen mit den geringsten Rechten gewähren, da sie für alle AWS Kunden verfügbar sind. Wir empfehlen Ihnen, die Berechtigungen weiter zu reduzieren, indem Sie kundenverwaltete Richtlinien definieren, die speziell auf Ihre Anwendungsfälle zugeschnitten sind.

Sie können die in AWS verwalteten Richtlinien definierten Berechtigungen nicht ändern. Wenn die in einer AWS verwalteten Richtlinie definierten Berechtigungen AWS aktualisiert werden, wirkt sich das Update auf alle Prinzidentitäten (Benutzer, Gruppen und Rollen) aus, denen die Richtlinie zugeordnet ist. AWS aktualisiert eine AWS verwaltete Richtlinie höchstwahrscheinlich, wenn eine neue Richtlinie eingeführt AWS -Service wird oder neue API Operationen für bestehende Dienste verfügbar werden.

Weitere Informationen finden Sie im IAMBenutzerhandbuch unter AWS Verwaltete Richtlinien.

AWS verwaltete Richtlinie: A mazonMQService RolePolicy

Sie können nichts `AmazonMQServiceRolePolicy` an Ihre IAM Entitäten anhängen. Diese Richtlinie ist einer serviceverknüpften Rolle zugeordnet, die Amazon MQ erlaubt, Aktionen in Ihrem Namen durchzuführen. Weitere Informationen zu dieser Berechtigungsrichtlinie und den Aktionen, die Amazon MQ ausführen kann, finden Sie unterthe section called “Serviceverknüpfte Rollenberechtigungen für Amazon MQ”.

Amazon MQ MQ-Updates für AWS verwaltete Richtlinien

Sehen Sie sich Details zu Aktualisierungen der AWS verwalteten Richtlinien für Amazon MQ an, seit dieser Service begonnen hat, diese Änderungen zu verfolgen. Um automatische Benachrichtigungen über Änderungen an dieser Seite zu erhalten, abonnieren Sie den RSS Feed auf der Amazon MQ MQ-Dokumentverlaufsseite.

Änderung	Beschreibung	Datum
Amazon MQ hat mit der Verfolgung von Änderungen begonnen	Amazon MQ hat damit begonnen, Änderungen an seinen AWS verwalteten Richtlinien nachzuverfolgen.	5. Mai 2021

Verwendung von serviceverknüpften Rollen für Amazon MQ

Amazon MQ verwendet AWS Identity and Access Management (IAM) serviceverknüpfte Rollen. Eine serviceverknüpfte Rolle ist ein spezieller Typ einer IAM-Rolle, die direkt mit Amazon MQ verknüpft ist. Serviceverknüpfte Rollen werden von Amazon MQ vordefiniert und schließen alle Berechtigungen ein, die der Service zum Aufrufen anderer AWS-Services in Ihrem Namen erfordert.

Eine serviceverknüpfte Rolle macht die Einrichtung von Amazon MQ einfacher, da Sie die erforderlichen Berechtigungen nicht manuell hinzufügen müssen. Amazon MQ definiert die Berechtigungen seiner serviceverknüpften Rollen, und sofern nicht anders definiert, kann nur Amazon MQ seine Rollen übernehmen. Die definierten Berechtigungen umfassen die Vertrauens- und Berechtigungsrichtlinie. Diese Berechtigungsrichtlinie kann keinen anderen IAM-Entitäten zugewiesen werden.

Sie können eine serviceverknüpfte Rolle erst löschen, nachdem ihre verwandten Ressourcen gelöscht wurden. Dies schützt Ihre Amazon MQ-Ressourcen, da Sie nicht versehentlich die Zugriffsberechtigung für die Ressourcen entfernen können.

Informationen zu anderen Services, die serviceverknüpfte Rollen unterstützen, finden Sie unter AWS-Services, die mit IAM funktionieren. Suchen Sie nach den Services, für die Yes (Ja) in der Spalte Serviceverknüpfte Rolle angegeben ist. Wählen Sie über einen Link Ja aus, um die Dokumentation zu einer servicegebundenen Rolle für diesen Service anzuzeigen.

Serviceverknüpfte Rollenberechtigungen für Amazon MQ

Amazon MQ verwendet die serviceverknüpfte Rolle namensAWSServiceRoleForAmazonMQ – Amazon MQ verwendet diese serviceverknüpfte Rolle, um AWS-Dienstleistungen in Ihrem Namen.

Die servicegebundene Rolle AWSServiceRoleForRDS vertraut den folgenden Services, die diese Rolle übernehmen:

- `mq.amazonaws.com`

Amazon MQ verwendet die Berechtigungsrichtlinie `AmazonMQServiceRolePolicy`, die serviceverknüpfte Rolle AWSServiceRoleForAmazonMQ enthält, um die folgenden Aktionen für die angegebenen Ressourcen durchzuführen:

- Aktion: `ec2:CreateVpcEndpoint` auf der `vpc`-Ressource.
- Aktion: `ec2:CreateVpcEndpoint` auf der `subnet`-Ressource.
- Aktion: `ec2:CreateVpcEndpoint` auf der `security-group`-Ressource.
- Aktion: `ec2:CreateVpcEndpoint` auf der `vpc-endpoint`-Ressource.
- Aktion: `ec2:DescribeVpcEndpoints` auf der `vpc`-Ressource.
- Aktion: `ec2:DescribeVpcEndpoints` auf der `subnet`-Ressource.
- Aktion: `ec2:CreateTags` auf der `vpc-endpoint`-Ressource.
- Aktion: `logs:PutLogEvents` auf der `log-group`-Ressource.
- Aktion: `logs:DescribeLogStreams` auf der `log-group`-Ressource.
- Aktion: `logs:DescribeLogGroups` auf der `log-group`-Ressource.
- Aktion: `CreateLogStream` auf der `log-group`-Ressource.
- Aktion: `CreateLogGroup` auf der `log-group`-Ressource.

Wenn Sie einen Amazon-MQ-für-RabbitMQ-Broker erstellen, erlaubt die `AmazonMQServiceRolePolicy`-Berechtigungsrichtlinie Amazon MQ die Durchführung der folgenden Aufgaben in Ihrem Namen.

- Erstellen Sie einen Amazon-VPC-Endpunkt für den Broker mithilfe der von Ihnen bereitgestellten Amazon VPC, des Subnetzes und der Sicherheitsgruppe. Sie können den für Ihren Broker erstellten Endpunkt verwenden, um sich über die RabbitMQ-Verwaltungskonsole, die Verwaltungs-API oder programmatisch mit dem Broker zu verbinden.
- Erstellen Sie Protokollgruppen und veröffentlichen Sie Broker-Protokolle in 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/*"
            ]
        }
    ]
```

```
}
```

Sie müssen Berechtigungen konfigurieren, damit eine juristische Stelle von IAM (z. B. Benutzer, Gruppe oder Rolle) eine servicegebundene Rolle erstellen, bearbeiten oder löschen kann. Weitere Informationen finden Sie unter servicegebundene Rollenberechtigungen im IAM-Benutzerhandbuch.

Erstellen einer serviceverknüpften Rolle für Amazon MQ

Sie müssen eine servicegebundene Rolle nicht manuell erstellen. Wenn Sie das erste Mal einen Broker erstellen, erstellt Amazon MQ eine serviceverknüpfte Rolle, die AWS-Services in Ihrem Namen aufruft. Alle nachfolgenden Broker, die Sie erstellen, verwenden dieselbe Rolle, und es wird keine neue Rolle erstellt.

> **Important**
>
> Diese serviceverknüpfte Rolle kann in Ihrem Konto erscheinen, wenn Sie eine Aktion in einem anderen Service abgeschlossen haben, der die von dieser Rolle unterstützten Funktionen verwendet. Weitere Informationen finden Sie unter Eine neue Rolle ist in meinem IAM-Konto erschienen.

Wenn Sie diese serviceverknüpfte Rolle löschen und sie dann erneut erstellen müssen, können Sie dasselbe Verfahren anwenden, um die Rolle in Ihrem Konto neu anzulegen.

Sie können auch die IAM-Konsole verwenden, um eine serviceverknüpfte Rolle mit dem Anwendungsfall Amazon MQ zu erstellen. Erstellen Sie in der AWS CLI oder der AWS-API eine servicegebundene Rolle mit dem Servicenamen `mq.amazonaws.com`. Weitere Informationen finden Sie unter Erstellen einer serviceverknüpften Rolle im IAM-Benutzerhandbuch. Wenn Sie diese servicegebundene Rolle löschen, können Sie mit demselben Verfahren die Rolle erneut erstellen.

Bearbeiten einer serviceverknüpften Rolle für Amazon MQ

Amazon MQ erlaubt es Ihnen nicht, die serviceverknüpfte Rolle AWSServiceRoleForAmazonMQ zu bearbeiten. Sie können jedoch die Beschreibung der Rolle mit IAM bearbeiten. Weitere Informationen finden Sie unter Bearbeiten einer serviceverknüpften Rolle im IAM-Benutzerhandbuch.

Löschen einer serviceverknüpften Rolle für Amazon MQ

Wenn Sie eine Funktion oder einen Service, die bzw. der eine servicegebundene Rolle erfordert, nicht mehr benötigen, sollten Sie diese Rolle löschen. Auf diese Weise haben Sie keine ungenutzte

juristische Stelle, die nicht aktiv überwacht oder verwaltet wird. Sie müssen jedoch die Ressourcen für Ihre servicegebundene Rolle zunächst bereinigen, bevor Sie sie manuell löschen können.

> **Note**
>
> Wenn der Amazon MQ-Service die Rolle verwendet, wenn Sie versuchen, die Ressourcen zu löschen, schlägt der Löschvorgang möglicherweise fehl. Wenn das passiert, warten Sie einige Minuten und versuchen Sie es erneut.

So löschen Sie Amazon MQ-Ressourcen, die von der Rolle AWSServiceRoleForAWSLicenseManagerRole verwendet werden:

- Löschen Sie Ihre Amazon MQ -Broker mit demAWS Management Console, Amazon MQ CLI oder Amazon MQ API. Weitere Informationen zum Löschen von Brokern finden Sie unter ???.

So löschen Sie die servicegebundene Rolle mit IAM

Verwenden Sie die IAM-Konsole, die AWS CLI oder die AWS-API, um die serviceverknüpfte Rolle „AWSServiceRoleForIVSRecordToS3“ zu löschen. Weitere Informationen finden Sie unter Löschen einer servicegebundenen Rolle im IAM-Leitfaden

Unterstützte Regionen für Amazon MQ serviceverknüpfte Rollen

Amazon MQ unterstützt die Verwendung von serviceverknüpften Rollen in allen Regionen, in denen der Service verfügbar ist. Weitere Informationen finden Sie unter AWS-Regionen und Endpunkte.

Name der Region	Regions-ID	Amazon MQ Support
USA Ost (Nord-Virginia)	us-east-1	Ja
USA Ost (Ohio)	us-east-2	Ja
USA West (Nordkalifornien)	us-west-1	Ja
USA West (Oregon)	us-west-2	Ja
Asien-Pazifik (Mumbai)	ap-south-1	Ja
Asien-Pazifik (Osaka)	ap-northeast-3	Ja

Name der Region	Regions-ID	Amazon MQ Support
Asien-Pazifik (Seoul)	ap-northeast-2	Ja
Asien-Pazifik (Singapore)	ap-southeast-1	Ja
Asien-Pazifik (Sydney)	ap-southeast-2	Ja
Asien-Pazifik (Tokyo)	ap-northeast-1	Ja
Kanada (Zentral)	ca-central-1	Ja
Europa (Frankfurt)	eu-central-1	Ja
Europa (Ireland)	eu-west-1	Ja
Europa (London)	eu-west-2	Ja
Europa (Paris)	eu-west-3	Ja
Südamerika (São Paulo)	sa-east-1	Ja
AWS GovCloud (US)	us-gov-west-1	Nein

Fehlerbehebung für Amazon MQ-Identität und -Zugriff

Verwenden Sie die folgenden Informationen, um häufig auftretende Probleme zu diagnostizieren und zu beheben, die bei der Arbeit mit Amazon MQ und IAM auftreten können.

Themen

- Ich bin nicht autorisiert, eine Aktion in Amazon MQ auszuführen
- Ich bin nicht berechtigt, iam durchzuführen: PassRole
- Ich möchte Personen außerhalb meines AWS Kontos den Zugriff auf meine Amazon MQ MQ-Ressourcen ermöglichen

Ich bin nicht autorisiert, eine Aktion in Amazon MQ auszuführen

Wenn Ihnen AWS Management Console mitgeteilt wird, dass Sie nicht berechtigt sind, eine Aktion durchzuführen, müssen Sie sich an Ihren Administrator wenden, um Unterstützung zu erhalten. Ihr Administrator hat Ihnen Ihre Anmeldeinformationen zur Verfügung gestellt.

Der folgende Beispielfehler tritt auf, wenn der `mateojackson` Benutzer versucht, die Konsole zu verwenden, um Details zu einem *`widget`* hat aber keine `mq:`*`GetWidget`* Berechtigungen.

```
User: arn:aws:iam::123456789012:user/mateojackson is not authorized to perform:
 mq:GetWidget on resource: my-example-widget
```

In diesem Fall bittet Mateo seinen Administrator um die Aktualisierung seiner Richtlinien, um unter Verwendung der Aktion *`my-example-widget`* auf die Ressource `mq:`*`GetWidget`* zugreifen zu können.

Ich bin nicht berechtigt, iam durchzuführen: PassRole

Wenn Sie die Fehlermeldung erhalten, dass Sie nicht zur Ausführung der Aktion „`iam:PassRole`“ autorisiert sind, müssen Ihre Richtlinien aktualisiert werden, um eine Rolle an Amazon MQ übergeben zu können.

Einige AWS -Services ermöglichen es Ihnen, eine bestehende Rolle an diesen Dienst zu übergeben, anstatt eine neue Servicerolle oder eine dienstverknüpfte Rolle zu erstellen. Hierzu benötigen Sie Berechtigungen für die Übergabe der Rolle an den Dienst.

Der folgende Beispielfehler tritt auf, wenn ein IAM Benutzer mit dem Namen `marymajor` versucht, die Konsole zu verwenden, um eine Aktion in Amazon MQ auszuführen. Die Aktion erfordert jedoch, dass der Service über Berechtigungen verfügt, die durch eine Servicerolle gewährt werden. Mary besitzt keine Berechtigungen für die Übergabe der Rolle an den Dienst.

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

In diesem Fall müssen die Richtlinien von Mary aktualisiert werden, um die Aktion `iam:PassRole` ausführen zu können.

Wenn Sie Hilfe benötigen, wenden Sie sich an Ihren AWS Administrator. Ihr Administrator hat Ihnen Ihre Anmeldeinformationen zur Verfügung gestellt.

Ich möchte Personen außerhalb meines AWS Kontos den Zugriff auf meine Amazon MQ MQ-Ressourcen ermöglichen

Sie können eine Rolle erstellen, die Benutzer in anderen Konten oder Personen außerhalb Ihrer Organisation für den Zugriff auf Ihre Ressourcen verwenden können. Sie können festlegen, wem die Übernahme der Rolle anvertraut wird. Für Dienste, die ressourcenbasierte Richtlinien oder Zugriffskontrolllisten (ACLs) unterstützen, können Sie diese Richtlinien verwenden, um Personen Zugriff auf Ihre Ressourcen zu gewähren.

Weitere Informationen dazu finden Sie hier:

- Informationen dazu, ob Amazon MQ diese Funktionen unterstützt, finden Sie unter So funktioniert Amazon MQ mit IAM.
- Informationen dazu, wie Sie Zugriff auf Ihre Ressourcen in AWS-Konten Ihrem Besitz gewähren können, finden Sie im Benutzerhandbuch unter Gewähren des Zugriffs IAM für einen Benutzer in einem anderen AWS-Konto , dem IAM Sie gehören.
- Informationen dazu, wie Sie Dritten Zugriff auf Ihre Ressourcen gewähren können AWS-Konten, finden Sie AWS-Konten im IAMBenutzerhandbuch unter Gewähren des Zugriffs für Dritte.
- Informationen dazu, wie Sie Zugriff über einen Identitätsverbund gewähren, finden Sie im Benutzerhandbuch unter Zugriff für extern authentifizierte Benutzer (Identitätsverbund). IAM
- Informationen zum Unterschied zwischen der Verwendung von Rollen und ressourcenbasierten Richtlinien für den kontenübergreifenden Zugriff finden Sie IAMim Benutzerhandbuch unter Kontoübergreifender Ressourcenzugriff. IAM

Compliance-Validierung für Amazon MQ

Externe Prüfer bewerten die Sicherheit und Konformität von Amazon MQ im Rahmen mehrerer AWS Compliance-Programme. Dazu gehörenSOC, PCIHIPAA, und andere.

Informationen darüber, ob AWS -Service ein AWS -Services in den Geltungsbereich bestimmter Compliance-Programme fällt, finden Sie unter Umfang nach Compliance-Programm AWS -Services unter . Wählen Sie dort das Compliance-Programm aus, an dem Sie interessiert sind. Allgemeine Informationen finden Sie unter AWS Compliance-Programme AWS .

Sie können Prüfberichte von Drittanbietern unter herunterladen AWS Artifact. Weitere Informationen finden Sie unter Berichte herunterladen unter .

Ihre Verantwortung für die Einhaltung der Vorschriften bei der Nutzung AWS -Services hängt von der Vertraulichkeit Ihrer Daten, den Compliance-Zielen Ihres Unternehmens und den geltenden Gesetzen und Vorschriften ab. AWS stellt die folgenden Ressourcen zur Verfügung, die Sie bei der Einhaltung der Vorschriften unterstützen:

- Schnellstartanleitungen zu Sicherheit und Compliance — In diesen Bereitstellungsleitfäden werden architektonische Überlegungen erörtert und Schritte für die Bereitstellung von Basisumgebungen beschrieben AWS , bei denen Sicherheit und Compliance im Mittelpunkt stehen.
- Architecting for HIPAA Security and Compliance on Amazon Web Services — In diesem Whitepaper wird beschrieben, wie Unternehmen Anwendungen erstellen HIPAA können, die AWS für sie in Frage kommen.

  > **Note**
  >
  > Nicht alle sind berechtigt AWS -Services . HIPAA Weitere Informationen finden Sie in der Referenz für HIPAA qualifizierte Dienste.

- AWS Ressourcen zur AWS von Vorschriften — Diese Sammlung von Arbeitsmappen und Leitfäden kann auf Ihre Branche und Ihren Standort zutreffen.
- AWS Leitfäden zur Einhaltung von Vorschriften für Kunden — Verstehen Sie das Modell der gemeinsamen Verantwortung aus dem Blickwinkel der Einhaltung von Vorschriften. In den Leitfäden werden die bewährten Verfahren zur Sicherung zusammengefasst AWS -Services und die Leitlinien für Sicherheitskontrollen in verschiedenen Frameworks (einschließlich des National Institute of Standards and Technology (NIST), des Payment Card Industry Security Standards Council (PCI) und der International Organization for Standardization (ISO)) zusammengefasst.
- Evaluierung von Ressourcen anhand von Regeln im AWS Config Entwicklerhandbuch — Der AWS Config Service bewertet, wie gut Ihre Ressourcenkonfigurationen den internen Praktiken, Branchenrichtlinien und Vorschriften entsprechen.
- AWS Security Hub— Dies AWS -Service bietet einen umfassenden Überblick über Ihren internen Sicherheitsstatus. AWS Security Hub verwendet Sicherheitskontrollen, um Ihre AWS -Ressourcen zu bewerten und Ihre Einhaltung von Sicherheitsstandards und bewährten Methoden zu überprüfen. Eine Liste der unterstützten Services und Kontrollen finden Sie in der Security-Hub-Steuerungsreferenz.
- Amazon GuardDuty — Dies AWS -Service erkennt potenzielle Bedrohungen für Ihre Workloads AWS-Konten, Container und Daten, indem es Ihre Umgebung auf verdächtige und böswillige Aktivitäten überwacht. GuardDuty kann Ihnen helfen, verschiedene Compliance-Anforderungen zu

erfüllen PCIDSS, z. B. durch die Erfüllung der Anforderungen zur Erkennung von Eindringlingen, die in bestimmten Compliance-Frameworks vorgeschrieben sind.

- AWS Audit Manager— Auf diese AWS -Service Weise können Sie Ihre AWS Nutzung kontinuierlich überprüfen, um das Risikomanagement und die Einhaltung von Vorschriften und Industriestandards zu vereinfachen.

Ausfallsicherheit bei Amazon MQ

Im Zentrum der globalen AWS-Infrastruktur stehen die AWS-Regionen und Availability Zones (Verfügbarkeitszonen, AZs). AWS Regionen stellen mehrere physisch getrennte und isolierte Availability Zones bereit, die über hoch redundante Netzwerke mit niedriger Latenz und hohen Durchsätzen verbunden sind. Mithilfe von Availability Zones können Sie Anwendungen und Datenbanken erstellen und ausführen, die automatisch Failover zwischen Zonen ausführen, ohne dass es zu Unterbrechungen kommt. Availability Zones sind besser hoch verfügbar, fehlertoleranter und skalierbarer als herkömmliche Infrastrukturen mit einem oder mehreren Rechenzentren.

Weitere Informationen über AWS-Regionen und -Availability Zones finden Sie unter Globale AWS-Infrastruktur.

Infrastruktursicherheit in Amazon MQ

Als verwalteter Service ist er durch AWS globale Netzwerksicherheit geschützt. Informationen zu AWS Sicherheitsdiensten und zum AWS Schutz der Infrastruktur finden Sie unter AWS Cloud-Sicherheit. Informationen zum Entwerfen Ihrer AWS Umgebung unter Verwendung der bewährten Methoden für die Infrastruktursicherheit finden Sie unter Infrastructure Protection in Security Pillar AWS Well-Architected Framework.

Sie verwenden AWS veröffentlichte API Aufrufe für den Zugriff über das Netzwerk. Kunden müssen Folgendes unterstützen:

- Sicherheit auf Transportschicht (TLS). Wir benötigen TLS 1.2 und empfehlen TLS 1.3.
- Cipher-Suites mit perfekter Vorwärtsgeheimhaltung (PFS) wie (Ephemeral Diffie-Hellman) oder DHE (Elliptic Curve Ephemeral Diffie-Hellman). ECDHE Die meisten modernen Systeme wie Java 7 und höher unterstützen diese Modi.

Darüber hinaus müssen Anfragen mithilfe einer Zugriffsschlüssel-ID und eines geheimen Zugriffsschlüssels, der einem Prinzipal zugeordnet ist, signiert werden. IAM Alternativ können Sie mit

AWS Security Token Service (AWS STS) temporäre Sicherheitsanmeldeinformationen erstellen, um die Anforderungen zu signieren.

Best Practices für die Sicherheit in Amazon MQ

Die folgenden Entwurfsmuster können die Sicherheit Ihres Amazon MQ-Broker verbessern.

Themen

- Broker ohne öffentlichen Zugriff bevorzugen
- Immer eine Autorisierungszuordnung konfigurieren
- Unnötige Protokolle mit VPC-Sicherheitsgruppen bockieren

Weitere Informationen dazu, wie Amazon MQ Ihre Daten verschlüsselt, sowie eine Liste der unterstützten Protokolle finden Sie unterDatenschutz.

Broker ohne öffentlichen Zugriff bevorzugen

Für Broker ohne öffentliche Zugänglichkeit ist kein Zugriff von außerhalb Ihrer VPC möglich. Dies reduziert die Anfälligkeit Ihres Brokers für DDoS-Angriffe (Distributed Denial of Service) aus dem öffentlichen Internet ganz wesentlich. Weitere Informationen finden Sie unter Zugriff auf die Amazon MQ Broker-Webkonsole ohne öffentlichen Zugriff in diesem Handbuch und unter Vorbereitung auf DDoS-Angriffe durch die Verringerung der Angriffsfläche im AWS-Blog zur Sicherheit.

Immer eine Autorisierungszuordnung konfigurieren

Da für ActiveMQ standardmäßig keine Autorisierungszuordnung konfiguriert ist, kann jeder authentifizierte Benutzer eine Aktion auf dem Broker ausführen. Daher ist es eine bewährte Methode, Berechtigungen nach Gruppe einzuschränken. Weitere Informationen finden Sie unter `authorizationEntry`.

> **Important**
>
> Wenn Sie eine Autorisierungszuordnung angeben, die die`activemq-webconsole`können Sie die ActiveMQ Webkonsole nicht verwenden, da die Gruppe nicht berechtigt ist, Nachrichten an den Amazon MQ -Broker zu senden oder von ihm Nachrichten zu empfangen.

Unnötige Protokolle mit VPC-Sicherheitsgruppen bockieren

Um die Sicherheit zu erhöhen, sollten Sie die Verbindungen von unnötigen Protokolle und Ports, indem Sie Ihre Amazon VPC-Sicherheitsgruppe ordnungsgemäß konfigurieren. Beispielsweise können Sie zur Einschränkung des Zugriffs auf die meisten Protokolle bei gleichzeitiger Gewährung des Zugriffs zu OpenWire und zur Webkonsole den Zugriff lediglich auf 61617 und 8162 erlauben. Dies begrenzt Gefahren durch die Blockierung von Protokollen, die Sie nicht verwenden, während OpenWire und die Webkonsole normal funktionieren können.

Erlauben Sie nur die Protokoll-Ports, die Sie verwenden.

- AMQP: 5671
- MQTT: 8883
- OpenWire: 61617
- STOMP: 61614
- WebSocket: 61619

Weitere Informationen finden Sie unter:

- Zusätzliche Amazon MQ-Broker-Einstellungen konfigurieren
- Sicherheitsgruppen für Ihre VPC
- Standardsicherheitsgruppe für Ihre VPC
- Arbeiten mit Sicherheitsgruppen

Überwachen und Protokollieren von Amazon MQ-Brokern

Die Überwachung ist ein wichtiger Bestandteil der Aufrechterhaltung der Zuverlässigkeit, Verfügbarkeit und Leistung Ihrer Lösungen. AWS Sie sollten Überwachungsdaten aus allen Teilen Ihrer AWS Lösung sammeln, damit Sie einen etwaigen Ausfall an mehreren Stellen leichter debuggen können. AWS bietet verschiedene Tools zur Überwachung Ihrer Amazon MQ MQ-Ressourcen und zur Reaktion auf potenzielle Vorfälle:

Sie können CloudWatch es verwenden, um Metriken für Ihren Amazon MQ-Broker anzuzeigen und zu analysieren. Sie können Ihre Broker-Metriken über die CloudWatch Konsole, den oder den CloudWatch AWS CLI anzeigen AWS CLI und analysieren. CloudWatch Die Metriken für Amazon MQ werden automatisch vom Broker abgefragt und dann auf CloudWatch jede Minute übertragen. CloudWatch Überwacht bei ActiveMQ-Brokern nur die ersten 1000 Ziele.. CloudWatch Überwacht bei RabbitMQ-Brokern nur die ersten 500 Ziele, sortiert nach der Anzahl der Verbraucher.

Eine vollständige Liste der Amazon MQ-Metriken finden Sie unter Verfügbare CloudWatch Metriken Amazon MQ für ActiveMQ-Broker.

Informationen zum Erstellen eines CloudWatch Alarms für eine Metrik finden Sie unter CloudWatch Alarm erstellen oder bearbeiten im CloudWatch Amazon-Benutzerhandbuch.

Zugreifen auf CloudWatch Metriken für Amazon MQ

Sie können mit AWS Management Console, AWS CLI und auf CloudWatch Metriken zugreifenAPI.

Möglicherweise möchten Sie auf CloudWatch Metriken zugreifen, ohne das zu verwenden AWS Management Console.

Verwenden Sie den `get-metric-statistics` Befehl, um mit dem AWS CLI auf Amazon MQ-Metriken zuzugreifen. Weitere Informationen finden Sie unter Get Statistics for a Metric im CloudWatch Amazon-Benutzerhandbuch.

Verwenden Sie die `GetMetricStatistics` Aktion, um mithilfe von auf Amazon MQ-Metriken zuzugreifen. CloudWatch API Weitere Informationen finden Sie unter Get Statistics for a Metric im CloudWatch Amazon-Benutzerhandbuch.

Zugreifen auf CloudWatch Metriken mit dem AWS Management Console

Das folgende Beispiel zeigt Ihnen, wie Sie mit dem auf CloudWatch Metriken für Amazon MQ zugreifen können. AWS Management Console Wenn Sie bereits bei der Amazon MQ-Konsole angemeldet sind, wählen Sie auf der Seite mit den Broker-Details die Optionen Aktionen, Metriken anzeigen aus. CloudWatch

1. Melden Sie sich bei der Konsole an. CloudWatch
2. Wählen Sie im Navigationsbereich Metriken aus.
3. Wählen Sie den Namespace der AmazonMQ-Metrik aus.
4. Wählen Sie eine der folgenden Metrik-Dimensionen aus:

   - Broker-Metriken
   - Warteschlangenmetriken nach Broker
   - Themenbezogene Metriken nach Broker

   In diesem Beispiel wird Broker-Metriken ausgewählt.
5. Sie können Ihre Amazon MQ-Metriken jetzt analysieren:

   - Verwenden Sie die Spaltenüberschrift, um die Metriken zu sortieren.
   - Um die Metrik grafisch darzustellen, müssen Sie das Kontrollkästchen neben der Metrik aktivieren.
   - Um nach Metrik zu filtern, müssen Sie den Metriknamen und anschließend Zur Suche hinzufügen auswählen.

Verfügbare CloudWatch Metriken Amazon MQ für ActiveMQ-Broker

Amazon MQ für ActiveMQ Metriken

Metrik	Einheit	Beschreibung
`AmqpMaximumConnections`	Anzahl	Die maximale Anzahl von Clients, mit denen Sie eine Verbindung zu Ihrem Broker

Metrik	Einheit	Beschreibung
		herstellen könnenAMQP. Weitere Informationen zu Verbindungskontingenten finden Sie unter Quotas in Amazon MQ.
`BurstBalance`	Prozent	Der Prozentsatz der verbleibenden Burst-Credits auf dem EBS Amazon-Volumen, der für die Beibehaltung von Nachrichtendaten für durchsatzoptimierte Broker verwendet wird. Wenn dieses Guthaben Null erreicht, verringert sich das von Amazon IOPS bereitgestellte EBS Volumen, bis das Burst-Guthaben wieder aufgefüllt ist. Weitere Informationen zur Funktionsweise von Burst Balances in Amazon EBS finden Sie unter I/O Credits und Burst Performance.

Metrik	Einheit	Beschreibung
`CpuCreditBalance`	Guthaben (v CPU Minuten)	**Important** Diese Metrik ist nur für den Broker-Instance-Typ `mq.t2.micro` verfügbar. CPUKreditkennzahlen sind nur in Intervallen von fünf Minuten verfügbar. Die Anzahl der verdienten CPU Credits, die eine Instance seit ihrem Start oder Start gesammelt hat (einschließlich der Anzahl der Start-Credits). Das Guthaben steht der Broker-Instance zur Verfügung, um es für Bursts auszugeben, die über die Basisnutzung hinausgehen. CPU Guthaben werden nach ihrem Erwerb im Guthabenkonto angesammelt und nach ihrer Verwendung daraus entfernt. Das Guthabenkonto hat eine Obergrenze. Nach Erreichen dieser Grenze werden neu verdiente Guthaben verworfen.

Metrik	Einheit	Beschreibung
CpuUtilization	Prozent	Der Prozentsatz der zugewiesenen EC2 Amazon-Recheneinheiten, die der Broker derzeit verwendet.
CurrentConnectionsCount	Anzahl	Die derzeitige Anzahl der aktiven Verbindungen auf dem aktuellen Broker.
EstablishedConnectionsCount	Anzahl	Die Gesamtzahl der aktiven und inaktiven Verbindungen, die auf dem Broker hergestellt wurden.
HeapUsage	Prozent	Der Prozentsatz des JVM ActiveMQ-Speicherlimits, den der Broker derzeit verwendet.
InactiveDurableTopicSubscribersCount	Anzahl	Die Anzahl der inaktiven dauerhaften Abonnenten des Themas, bis maximal 2000.
JobSchedulerStorePercentUsage	Prozent	Der Anteil des Festplatt enspeichers, der vom Speicher des Aufgaben-Schedulers belegt wird.
JournalFilesForFastRecovery	Anzahl	Die Anzahl der Journaldateien, die nach einem sauberen Shutdown erneut abgespielt werden.
JournalFilesForFullRecovery	Anzahl	Die Anzahl der Journaldateien, die nach einem unsauberen Shutdown erneut abgespielt werden.

Metrik	Einheit	Beschreibung
`MqttMaximumConnections`	Anzahl	Die maximale Anzahl von Clients, mit denen Sie eine Verbindung zu Ihrem Broker herstellen können. MQTT Weitere Informationen zu Verbindungskontingenten finden Sie unter Quotas in Amazon MQ.
`NetworkConnectorConnectionCount`	Anzahl	Die Anzahl der mit dem Broker verbundenen Knoten in einem Netzwerk von Brokern, die NetworkConnector.
`NetworkIn`	Bytes	Das Volumen des eingehenden Datenverkehrs für den Broker.
`NetworkOut`	Bytes	Das Volumen des ausgehenden Datenverkehrs für den Broker.
`OpenTransactionCount`	Anzahl	Die Gesamtzahl der in Bearbeitung befindlichen Transaktionen.
`OpenwireMaximumConnections`	Anzahl	Die maximale Anzahl von Clients, über die Sie eine Verbindung zu Ihrem Broker herstellen können OpenWire. Weitere Informationen zu Verbindungskontingenten finden Sie unter Quotas in Amazon MQ.

Metrik	Einheit	Beschreibung
StompMaximumConnections	Anzahl	Die maximale Anzahl von Clients, mit denen Sie eine Verbindung zu Ihrem Broker herstellen könnenSTOMP. Weitere Informationen zu Verbindungskontingenten finden Sie unter Quotas in Amazon MQ.
StorePercentUsage	Prozent	Der vom Speicherlimit verwendete Prozentsatz. Wenn dieser 100 erreicht, lehnt der Broker Nachrichten ab.
TempPercentUsage	Prozent	Der Anteil des verfügbaren temporären Speichers, der von nicht persistenten Nachrichten verwendet wird.
TotalConsumerCount	Anzahl	Die Gesamtzahl der Nachrichtennutzer, die Ziele auf dem aktuellen Broker abonniert haben.
TotalMessageCount	Anzahl	Die Anzahl der auf dem Broker gespeicherten Nachrichten.
TotalProducerCount	Anzahl	Die Gesamtzahl der Nachrichtenproduzenten, die auf Zielen auf dem aktuellen Broker aktiv sind.
VolumeReadOps	Anzahl	Die Anzahl der Lesevorgänge, die auf dem EBS Amazon-Volume ausgeführt wurden.

Metrik	Einheit	Beschreibung
`VolumeWriteOps`	Anzahl	Die Anzahl der Schreibvorgänge, die auf dem EBS Amazon-Volume ausgeführt wurden.
`WsMaximumConnections`	Anzahl	Die maximale Anzahl von Clients, mit denen Sie eine Verbindung zu Ihrem Broker herstellen können WebSocket. Weitere Informationen zu Verbindungskontingenten finden Sie unter Quotas in Amazon MQ.

Dimensionen für ActiveMQ-Broker-Metriken

Dimension	Beschreibung
`Broker`	Der Name des Brokers **Note** Ein Broker mit einer einzigen Instance hat das Suffix -1. Ein aktiver/Standby-Broker für hohe Verfügbarkeit hat die Suffixe -1 und -2 für sein redundantes Paar.

ActiveMQ-Ziel-Metriken (Warteschlange und Thema)

Important

Die folgenden Kennzahlen beinhalten Zählungen pro Minute für den CloudWatch Abfragezeitraum.

- `EnqueueCount`
- `ExpiredCount`
- `DequeueCount`
- `DispatchCount`
- `InFlightCount`

`EnqueueCount`Hat beispielsweise in einem CloudWatch Zeitraum von fünf Minuten fünf Zählwerte, jeweils für einen einminütigen Teil des Zeitraums. Die Statistiken `Minimum` und `Maximum` bieten den niedrigsten und höchsten Wert pro Minute während des angegebenen Zeitraums.

Metrik	Einheit	Beschreibung
`ConsumerCount`	Anzahl	Die Anzahl der Verbraucher, die das Ziel abonniert haben.
`EnqueueCount`	Anzahl	Die Anzahl der Nachrichten, die zum Ziel gesendet werden, pro Minute.
`EnqueueTime`	Zeit (Millisekunden)	Die end-to-end Latenz zwischen dem Eintreffen einer Nachricht bei einem Broker und ihrer Zustellung an einen Verbraucher.

Metrik	Einheit	Beschreibung
		Note `EnqueueTime` misst weder die end-to-end Latenz vom Senden einer Nachricht durch einen Hersteller bis zum Eingang beim Broker noch die Latenz vom Empfang einer Nachricht durch einen Broker bis zur Bestätigung durch den Broker. Vielmehr ist `EnqueueTime` die Anzahl der Millisekunden ab dem Zeitpunkt, an dem eine Nachricht vom Broker empfangen wird, bis sie erfolgreich an einen Verbraucher übermittelt wird.
`ExpiredCount`	Anzahl	Die Anzahl der Nachrichten pro Minute, die nicht übermittelt werden konnten, da sie abgelaufen sind.
`DispatchCount`	Anzahl	Die Anzahl der Nachrichten pro Minute, die an Verbraucher gesendet wurden.

Metrik	Einheit	Beschreibung
`DequeueCount`	Anzahl	Die Anzahl der Nachrichten, die von Verbrauchern bestätigt wurden.
`InFlightCount`	Anzahl	Die Anzahl der an Verbrauch er gesendeten Nachrichten, die nicht bestätigt wurden.
`ReceiveCount`	Anzahl	Die Anzahl der Nachrichten, die vom Remote-Broker für einen Duplex-Netzwerk-Co nnector empfangen wurden.
`MemoryUsage`	Prozent	Der Anteil am maximalen Arbeitsspeicher, der vom Ziel derzeit genutzt wird.
`ProducerCount`	Anzahl	Die Anzahl der Produzenten für das Ziel.
`QueueSize`	Anzahl	Die Anzahl der Nachrichten in der Warteschlange. **Important** Die Metrik gilt nur für Warteschlangen.
`TotalEnqueueCount`	Anzahl	Die Gesamtzahl der Nachricht en, die an den Broker gesendet wurden.
`TotalDequeueCount`	Anzahl	Die Gesamtanzahl der Nachrichten, die von Clients verwendet wurden.

> **Note**
>
> Die Metriken `TotalEnqueueCount` und `TotalDequeueCount` enthalten Nachrichten zu Beratungsthemen. Weitere Informationen zu Nachrichten zu Beratungsthemen finden Sie in der ActiveMQ-Dokumentation.

Dimensionen für ActiveMQ Ziel-Metriken (Warteschlange und Thema)

Dimension	Beschreibung
`Broker`	Der Name des Brokers. **Note** Ein Broker mit einer einzigen Instance hat das Suffix `-1`. Ein aktiver/Standby-Broker für hohe Verfügbarkeit hat die Suffixe `-1` und `-2` für sein redundantes Paar.
`Topic` oder `Queue`	Der Name des Themas oder der Warteschlange.
`NetworkConnector`	Der Name des Netzwerk-Connectors.

Verfügbare CloudWatch Metriken für Amazon MQ für RabbitMQ-Broker

RabbitMQ-Broker-Metriken

Metrik	Einheit	Beschreibung
`ExchangeCount`	Anzahl	Die Gesamtzahl der auf dem Broker konfigurierten Börsen.

Metrik	Einheit	Beschreibung
`QueueCount`	Anzahl	Die Gesamtanzahl der auf dem Broker konfigurierten Warteschlangen.
`ConnectionCount`	Anzahl	Die Gesamtzahl der auf dem Broker hergestellt wurden.
`ChannelCount`	Anzahl	Die Gesamtzahl der auf dem Broker festgelegten Kanäle.
`ConsumerCount`	Anzahl	Die Gesamtzahl der Verbraucher, die mit dem Broker verbunden sind.
`MessageCount`	Anzahl	Die Gesamtzahl der Nachrichten in der Warteschlange. **Note** Die produzierte Anzahl ist die Gesamtsumme der bereitgestellten und unerkannten Nachrichten auf dem Broker.
`MessageReadyCount`	Anzahl	Die Gesamtzahl der bereitgestellten Nachrichten in den Warteschlangen.
`MessageUnacknowledgedCount`	Anzahl	Die Gesamtzahl der nicht bestätigten Nachrichten in den Warteschlangen.

Metrik	Einheit	Beschreibung
`PublishRate`	Anzahl	Die Rate, mit der Nachrichten an den Broker veröffentlicht werden. Die erzeugte Zahl stellt die Anzahl der Nachrichten pro Sekunde zum Zeitpunkt der Probenahme dar.
`ConfirmRate`	Anzahl	Die Rate, mit der der RabbitMQ-Server veröffent lichte Nachrichten bestätigt. Sie können diese Metrik mit `PublishRate` vergleichen, um besser zu verstehen, wie Ihr Broker funktioniert. Die erzeugte Zahl stellt die Anzahl der Nachrichten pro Sekunde zum Zeitpunkt der Probenahme dar.
`AckRate`	Anzahl	Die Rate, mit der Nachricht en von den Verbrauchern anerkannt werden. Die erzeugte Zahl stellt die Anzahl der Nachrichten pro Sekunde zum Zeitpunkt der Probenahme dar.

Metrik	Einheit	Beschreibung
SystemCpuUtilization	Prozent	Der Prozentsatz der zugewiesenen EC2 Amazon-Recheneinheiten, die der Broker derzeit verwendet. Bei Cluster-Bereitstellungen stellt dieser Wert das Aggregat der entsprechenden Metrikwerte aller drei RabbitMQ-Knoten dar.
RabbitMQMemLimit	Bytes	Das RAM Limit für einen RabbitMQ-Broker. Bei Cluster-Bereitstellungen stellt dieser Wert das Aggregat der entsprechenden Metrikwerte aller drei RabbitMQ-Knoten dar.
RabbitMQMemUsed	Bytes	Das von einem RAM RabbitMQ-Broker genutzte Volumen. Bei Cluster-B ereitstellungen stellt dieser Wert das Aggregat der entsprechenden Metrikwerte aller drei RabbitMQ-Knoten dar.

Metrik	Einheit	Beschreibung
`RabbitMQDiskFreeLimit`	Bytes	Das Festplattenlimit für einen RabbitMQ-Broker. Bei Cluster-Bereitstellungen stellt dieser Wert das Aggregat der entsprechenden Metrikwerte aller drei RabbitMQ-Knoten dar. Diese Metrik unterscheidet sich je nach Instance-Größe. Weitere Informationen zu Amazon MQ Instance-Typen, finden Sie unter the section called “Instance-Typen von Amazon MQ für RabbitMQ”
`RabbitMQDiskFree`	Bytes	Das Gesamt-Volume des freien Speicherplatzes, der in einem RabbitMQ-Broker verfügbar ist. Wenn die Datenträgernutzung seinen Grenzwert überschreitet, blockiert der Cluster alle Herstellerverbindungen. Bei Cluster-Bereitstellungen stellt dieser Wert das Aggregat der entsprechenden Metrikwerte aller drei RabbitMQ-Knoten dar.

Metrik	Einheit	Beschreibung
RabbitMQFdUsed	Anzahl	Anzahl der verwendeten Datei-Deskriptoren Bei Cluster-Bereitstellungen stellt dieser Wert das Aggregat der entsprechenden Metrikwerte aller drei RabbitMQ-Knoten dar.
RabbitMQIOReadAverageTime	Anzahl	Die durchschnittliche Zeit (in Millisekunden), die RabbitMQf für einen Lesevorgang benötigt. Der Wert ist proportional zur Nachrichtengröße.
RabbitMQIOWriteAverageTime	Anzahl	Die durchschnittliche Zeit (in Millisekunden), die RabbitMQf ür einen Schreibvorgang benötigt. Der Wert ist proportional zur Nachrichtengröße.

Abmessungen für RabbitMQ-Broker-Metriken

Dimension	Beschreibung
Broker	Der Name des Brokers.

RabbitMQ-Knoten-Metriken

Metrik	Einheit	Beschreibung
SystemCpuUtilization	Prozent	Der Prozentsatz der zugewiesenen EC2 Amazon-

Metrik	Einheit	Beschreibung
		Recheneinheiten, die der Broker derzeit verwendet.
`RabbitMQMemLimit`	Bytes	Das RAM Limit für einen RabbitMQ-Knoten.
`RabbitMQMemUsed`	Bytes	Das Volumen, das von einem RAM RabbitMQ-Knoten verwendet wird. Wenn der Speicherverbrauch über das Limit hinausgeht, blockiert der Cluster alle Herstellerverbindungen.
`RabbitMQDiskFreeLimit`	Bytes	Das Festplattenlimit für einen RabbitMQ-Knoten. Diese Metrik unterscheidet sich je nach Instance-Größe. Weitere Informationen zu Amazon MQ Instance-Typen, finden Sie unter the section called “Instance-Typen von Amazon MQ für RabbitMQ”
`RabbitMQDiskFree`	Bytes	Das Gesamt-Volume des freien Speicherplatzes, der in einem RabbitMQ-Knoten verfügbar ist. Wenn die Datenträgernutzung seinen Grenzwert überschreitet, blockiert der Cluster alle Herstellerverbindungen.
`RabbitMQFdUsed`	Anzahl	Anzahl der verwendeten Datei-Deskriptoren

Abmessungen für RabbitMQ-Knotenmetriken

Dimension	Beschreibung
`Node`	Der Name des Knotens. **Note** Ein Knotenname besteht aus zwei Teilen: einem Präfix (üblicher weise `rabbit`) und einen Hostnamen . Zum Beispiel ist `rabbit@ip-10-0-0-230.us-west-2.compute.internal` ein Knotenname mit dem Präfix `rabbit` und dem Hostnamen `ip-10-0-0-230.us-west-2.compute.internal` .
`Broker`	Der Name des Brokers.

RabbitMQ-Warteschlangen-Metriken

Metrik	Einheit	Beschreibung
`ConsumerCount`	Anzahl	Die Anzahl der Verbraucher, die die Warteschlange abonniert haben.
`MessageReadyCount`	Anzahl	Die Anzahl der Nachrichten, die derzeit zugestellt werden können.
`MessageUnacknowledgedCount`	Anzahl	Die Anzahl der Nachrichten, für die der Server auf die Bestätigung wartet.

Metrik	Einheit	Beschreibung
`MessageCount`	Anzahl	Die Gesamtzahl für `MessageReadyCount` und `MessageUnacknowledgedCount` (auch als Warteschlangentiefe bezeichnet).

Dimensionen für RabbitMQ-Queue-Metriken

> **Note**
>
> Amazon MQ for RabbitMQ veröffentlicht keine Metriken für virtuelle Hosts und Warteschlangen, deren Namen Leerzeichen, Tabulatoren oder andere Nichtzeichen enthalten. ASCII
> Weitere Informationen zu Dimensionsnamen finden Sie unter Dimension in der CloudWatch APIAmazon-Referenz.

Dimension	Beschreibung
`Queue`	Der Name der Warteschlange.
`VirtualHost`	Der Name des virtuellen Hosts.
`Broker`	Der Name des Brokers.

Konfigurieren von Amazon MQ für RabbitMQ-Protokolle

Wenn Sie die CloudWatch Protokollierung für Ihre RabbitMQ-Broker aktivieren, verwendet Amazon MQ eine serviceverknüpfte Rolle, um allgemeine Protokolle zu veröffentlichen. CloudWatch Wenn beim Erstellen eines Brokers keine Rolle mit Amazon MQ vorhanden ist, erstellt Amazon MQ automatisch eine Rolle. Alle nachfolgenden RabbitMQ-Broker verwenden dieselbe servicebezogene Rolle für die Veröffentlichung von Protokollen. CloudWatch

Weitere Informationen zu dienstverknüpften Rollen finden Sie unter Verwenden von dienstbezogenen Rollen im Benutzerhandbuch.AWS Identity and Access Management Weitere Informationen darüber, wie Amazon MQ serviceverknüpfte Rollen verwendet, finden Sie unter the section called “Verwenden von servicegebundenen Rollen”.

Protokollieren Amazon MQ API MQ-Anrufen mit AWS CloudTrail

Amazon MQ ist in einen Service integriert AWS CloudTrail, der eine Aufzeichnung der Amazon MQ MQ-Aufrufe bereitstellt, die ein Benutzer, eine Rolle oder ein AWS Service tätigt. CloudTrail erfasst API Anrufe im Zusammenhang mit Amazon MQ-Brokern und Konfigurationen als Ereignisse, einschließlich Aufrufe von der Amazon MQ-Konsole und Codeaufrufen von Amazon MQ. APIs Weitere Informationen zu CloudTrail finden Sie im AWS CloudTrail Benutzerhandbuch.

> **Note**
>
> CloudTrail protokolliert keine API Aufrufe im Zusammenhang mit ActiveMQ-Vorgängen (z. B. Senden und Empfangen von Nachrichten) oder mit der ActiveMQ Web Console. Um Informationen zu ActiveMQ-Vorgängen zu protokollieren, können Sie Amazon MQ so konfigurieren, dass allgemeine Protokolle und Auditprotokolle in Amazon Logs veröffentlicht werden. CloudWatch

Anhand der CloudTrail gesammelten Informationen können Sie eine bestimmte Anfrage an einen Amazon MQAPI, die IP-Adresse des Anfragenden, die Identität des Anfragenden, Datum und Uhrzeit der Anfrage usw. identifizieren. Wenn Sie einen Trail konfigurieren, können Sie die kontinuierliche Übermittlung von CloudTrail Ereignissen an einen Amazon S3 S3-Bucket aktivieren. Wenn Sie keinen Trail konfigurieren, können Sie die neuesten Ereignisse im Ereignisverlauf in der CloudTrail Konsole einsehen. Weitere Informationen finden Sie unter Übersicht zum Erstellen eines Trails im AWS CloudTrail Benutzerhandbuch.

Amazon MQ MQ-Informationen in CloudTrail

Wenn Sie Ihr AWS Konto erstellen, CloudTrail ist aktiviert. Wenn eine unterstützte Amazon MQ MQ-Ereignisaktivität auftritt, wird sie zusammen mit anderen AWS Serviceereignissen in der CloudTrail Ereignishistorie als Ereignis aufgezeichnet. Sie können die neuesten Ereignisse für Ihr AWS -Konto anzeigen, durchsuchen und herunterladen. Weitere Informationen finden Sie im AWS CloudTrail Benutzerhandbuch unter Ereignisse mit CloudTrail Ereignisverlauf anzeigen.

Ein Trail ermöglicht CloudTrail die Übertragung von Protokolldateien an einen Amazon S3 S3-Bucket. Sie können einen Trail erstellen, um die Ereignisse in Ihrem AWS Konto fortlaufend aufzuzeichnen. Wenn du einen Trail mit dem erstellst AWS Management Console, gilt der Trail standardmäßig für alle AWS Regionen. Der Trail protokolliert Ereignisse aus allen AWS Regionen und übermittelt Protokolldateien an den angegebenen Amazon S3 S3-Bucket. Sie können auch andere AWS Dienste konfigurieren, um die in den CloudTrail Protokollen gesammelten Ereignisdaten weiter zu analysieren und darauf zu reagieren. Weitere Informationen finden Sie in folgenden Themen im AWS CloudTrail - Benutzerhandbuch:

- CloudTrail Unterstützte Dienste und Integrationen
- Konfiguration von SNS Amazon-Benachrichtigungen für CloudTrail
- Empfangen von CloudTrail Protokolldateien aus mehreren Regionen
- Empfangen von CloudTrail Protokolldateien von mehreren Konten

Amazon MQ unterstützt die Protokollierung sowohl der Anforderungsparameter als auch der Antworten für Folgendes APIs als Ereignisse in CloudTrail Protokolldateien:

- `CreateConfiguration`
- `DeleteBroker`
- `DeleteUser`
- `RebootBroker`
- `UpdateBroker`

> **Note**
>
> RebootBroker Protokolldateien werden protokolliert, wenn Sie den Broker neu starten. Während des Wartungsfensters wird der Dienst automatisch neu gestartet, und die RebootBroker Protokolldateien werden nicht protokolliert.

> **Important**
>
> Bei den folgenden `GET` APIs Methoden werden die Anforderungsparameter protokolliert, die Antworten jedoch geschwärzt:
>
> - `DescribeBroker`

- `DescribeConfiguration`
- `DescribeConfigurationRevision`
- `DescribeUser`
- `ListBrokers`
- `ListConfigurationRevisions`
- `ListConfigurations`
- `ListUsers`

Im Folgenden APIs werden die Parameter `data` und die `password` Anforderungsparameter durch Sternchen () verdeckt: `***`

- `CreateBroker` (POST)
- `CreateUser` (POST)
- `UpdateConfiguration` (PUT)
- `UpdateUser` (PUT)

Jedes Ereignis oder jeder Protokolleintrag enthält Informationen über den Ersteller der Anforderung. Mit diesen Informationen können Sie Folgendes bestimmen:

- Wurde die Anforderung mit Root- oder -Benutzeranmeldeinformationen ausgeführt?
- Wurde die Anforderung mit temporären Sicherheitsanmeldeinformationen für eine Rolle oder einen verbundenen Benutzer ausgeführt?
- Wurde die Anfrage von einem anderen AWS Dienst gestellt?

Weitere Informationen finden Sie unter CloudTrail userIdentity Element im AWS CloudTrail Benutzerhandbuch.

Beispiel für einen Amazon MQ-Protokolldateieintrag

Ein Trail ist eine Konfiguration, die die Übertragung von Ereignissen als Protokolldateien an den angegebenen Amazon S3 S3-Bucket ermöglicht. CloudTrail Protokolldateien enthalten einen oder mehrere Protokolleinträge.

Ein Ereignis stellt eine einzelne Anfrage aus einer beliebigen Quelle dar und umfasst Informationen über die Anfrage an einen Amazon MQAPI, die IP-Adresse des Anforderers, die Identität des Anfragenden, Datum und Uhrzeit der Anfrage usw.

Das folgende Beispiel zeigt einen CloudTrail Protokolleintrag für einen Anruf. `CreateBroker`API

> **Note**
>
> Da es CloudTrail sich bei Protokolldateien nicht um einen geordneten öffentlichen Stack-Trace handeltAPIs, listen sie Informationen nicht in einer bestimmten Reihenfolge auf.


```
{
    "eventVersion": "1.06",
    "userIdentity": {
        "type": "IAMUser",
        "principalId": "AKIAIOSFODNN7EXAMPLE",
        "arn": "arn:aws:iam::111122223333:user/AmazonMqConsole",
        "accountId": "111122223333",
        "accessKeyId": "AKIAI44QH8DHBEXAMPLE",
        "userName": "AmazonMqConsole"
    },
    "eventTime": "2018-06-28T22:23:46Z",
    "eventSource": "amazonmq.amazonaws.com",
    "eventName": "CreateBroker",
    "awsRegion": "us-west-2",
    "sourceIPAddress": "203.0.113.0",
    "userAgent": "PostmanRuntime/7.1.5",
    "requestParameters": {
        "engineVersion": "5.15.9",
        "deploymentMode": "ACTIVE_STANDBY_MULTI_AZ",
        "maintenanceWindowStartTime": {
            "dayOfWeek": "THURSDAY",
            "timeOfDay": "22:45",
            "timeZone": "America/Los_Angeles"
        },
        "engineType": "ActiveMQ",
        "hostInstanceType": "mq.m5.large",
        "users": [
            {
                "username": "MyUsername123",
                "password": "***",
```

```
                "consoleAccess": true,
                "groups": [
                    "admins",
                    "support"
                ]
            },
            {
                "username": "MyUsername456",
                "password": "***",
                "groups": [
                    "admins"
                ]
            }
        ],
        "creatorRequestId": "1",
        "publiclyAccessible": true,
        "securityGroups": [
            "sg-a1b234cd"
        ],
        "brokerName": "MyBroker",
        "autoMinorVersionUpgrade": false,
        "subnetIds": [
            "subnet-12a3b45c",
            "subnet-67d8e90f"
        ]
    },
    "responseElements": {
        "brokerId": "b-1234a5b6-78cd-901e-2fgh-3i45j6k178l9",
        "brokerArn": "arn:aws:mq:us-
east-2:123456789012:broker:MyBroker:b-1234a5b6-78cd-901e-2fgh-3i45j6k178l9"
    },
    "requestID": "a1b2c345-6d78-90e1-f2g3-4hi56jk7l890",
    "eventID": "a12bcd3e-fg45-67h8-ij90-12k34d5l16mn",
    "readOnly": false,
    "eventType": "AwsApiCall",
    "recipientAccountId": "111122223333"
}
```

Konfigurieren von Amazon MQ für ActiveMQ-Protokolle

Damit Amazon MQ Protokolle in Logs veröffentlichen kann, müssen Sie Ihrem Amazon MQ-Benutzer eine Berechtigung hinzufügen und außerdem eine ressourcenbasierte Richtlinie für Amazon MQ konfigurieren, bevor Sie den Broker erstellen oder neu starten. CloudWatch

> **Note**
>
> Wenn Sie Protokolle aktivieren und Nachrichten von der ActiveMQ-Webkonsole aus veröffentlichen, wird der Inhalt der Nachricht an die Protokolle gesendet CloudWatch und dort angezeigt.

Im Folgenden werden die Schritte zum Konfigurieren von CloudWatch Protokollen für Ihre ActiveMQ-Broker beschrieben.

Themen

Grundlegendes zur Struktur der Protokollierung von Protokollen CloudWatch

Sie können die allgemeine Protokollierung und die Audit-Protokollierung aktivieren, wenn Sie die erweiterten Broker-Einstellungen konfigurieren, einen Broker erstellen oder einen Broker bearbeiten.

Die allgemeine Protokollierung aktiviert die `INFO` Standardprotokollierungsebene (`DEBUG`Protokollierung wird nicht unterstützt) und veröffentlicht `activemq.log` in einer Protokollgruppe in Ihrem CloudWatch Konto. Die Protokollgruppe hat ein Format, das in etwa aussieht wie folgt:

```
/aws/amazonmq/broker/b-1234a5b6-78cd-901e-2fgh-3i45j6k178l9/general
```

Die Auditprotokollierung ermöglicht die Protokollierung von Verwaltungsaktionen, die mit JMX oder mithilfe der ActiveMQ Web Console durchgeführt wurden, und veröffentlicht `audit.log` sie in

einer Protokollgruppe in Ihrem CloudWatch Konto. Die Protokollgruppe hat ein Format, das in etwa aussieht wie folgt:

```
/aws/amazonmq/broker/b-1234a5b6-78cd-901e-2fgh-3i45j6k178l9/audit
```

Je nachdem, ob Sie einen Single-Instance-Broker oder einen Active-/Standby-Broker für hohe Verfügbarkeit verwenden, erstellt Amazon MQ entweder einen oder zwei Protokollstreams in jeder Protokollgruppe. Die Protokollstreams haben ein Format, das in etwa aussieht wie folgt:

```
activemq-b-1234a5b6-78cd-901e-2fgh-3i45j6k178l9-1.log
                activemq-b-1234a5b6-78cd-901e-2fgh-3i45j6k178l9-2.log
```

Die Suffixe `-1` und `-2` kennzeichnen einzelne Broker-Instances. Weitere Informationen finden Sie unter Arbeiten mit Protokollgruppen und Protokollstreams im Amazon CloudWatch Logs-Benutzerhandbuch.

Hinzufügen der **`CreateLogGroup`**-Berechtigung zu Ihrem Amazon-MQ-Benutzer

Damit Amazon MQ eine CloudWatch Logs-Protokollgruppe erstellen kann, müssen Sie sicherstellen, dass der Benutzer, der den Broker erstellt oder neu startet, über die entsprechenden Rechte verfügt. `logs:CreateLogGroup`

> **Important**
>
> Wenn Sie die `CreateLogGroup`-Berechtigung nicht zu Ihrem Amazon MQ-Benutzer hinzufügen, bevor der Benutzer den Broker erstellt oder neu startet, wird die Protokollgruppe nicht von Amazon MQ erstellt.

Die folgende IAMbeispielbasierte Richtlinie gewährt Benutzern, denen diese Richtlinie zugeordnet ist, die Erlaubnis. `logs:CreateLogGroup`

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

```
                "Resource": "arn:aws:logs:*:*:log-group:/aws/amazonmq/*"
            }
        ]
        }
```

> **Note**
>
> Hier bezieht sich der Begriff „Benutzer“ auf Benutzer und nicht auf Amazon-MQ-Benutzer, die erstellt werden, wenn ein neuer Broker konfiguriert wird. Weitere Informationen zur Einrichtung von Benutzern und zur Konfiguration von IAM Richtlinien finden Sie im Abschnitt Identity Management Overview des IAM Benutzerhandbuchs.

Weitere Informationen finden Sie `CreateLogGroup` in der Amazon CloudWatch API Logs-Referenz.

Konfigurieren einer ressourcenbasierten Richtlinie für Amazon MQ.

> **Important**
>
> Wenn Sie keine ressourcenbasierte Richtlinie für Amazon MQ konfigurieren, kann der Broker die Protokolle nicht in Logs veröffentlichen. CloudWatch

Damit Amazon MQ Protokolle in Ihrer Logs-Protokollgruppe veröffentlichen kann, konfigurieren Sie eine ressourcenbasierte Richtlinie, um Amazon MQ Zugriff auf die folgenden CloudWatch Logs-Aktionen zu gewähren: CloudWatch API

- `CreateLogStream`— Erstellt einen CloudWatch Logs-Log-Stream für die angegebene Protokollgruppe.
- `PutLogEvents`— Liefert Ereignisse in den angegebenen CloudWatch Log-Log-Stream.

Die folgende ressourcenbasierte Richtlinie gewährt Berechtigungen für `logs:CreateLogStream` und `logs:PutLogEvents` für. 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/*"
                                }
                            ]
                        }
```

Diese ressourcenbasierte Richtlinie muss mithilfe von konfiguriert werden, AWS CLI wie im folgenden Befehl gezeigt. Im Beispiel, ersetzen Sie *us-east-1* mit Ihren eigenen Informationen.

```
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**
>
> Da in diesem Beispiel das `/aws/amazonmq/` Präfix verwendet wird, müssen Sie die ressourcenbasierte Richtlinie nur einmal pro AWS Konto und Region konfigurieren.

Serviceübergreifende Confused-Deputy-Prävention

Das Confused-Deputy-Problem ist ein Sicherheitsproblem, bei dem eine juristische Stelle, die nicht über die Berechtigung zum Ausführen einer Aktion verfügt, eine privilegiertere juristische Stelle zwingen kann, die Aktion auszuführen. In AWS kann ein dienstübergreifendes Identitätswechsels zum Problem des verwirrten Stellvertreters führen. Ein dienstübergreifender Identitätswechsel kann auftreten, wenn ein Dienst (der Anruf-Dienst) einen anderen Dienst anruft (den aufgerufenen Dienst). Der Anruf-Dienst kann so manipuliert werden, dass er seine Berechtigungen verwendet, um auf die Ressourcen eines anderen Kunden zu reagieren, auf die er sonst nicht zugreifen dürfte. Um dies zu verhindern, AWS bietet Tools, mit denen Sie Ihre Daten für alle Dienste mit Dienstprinzipalen schützen können, denen Zugriff auf Ressourcen in Ihrem Konto gewährt wurde.

Wir empfehlen, die Kontextschlüssel `aws:SourceArn` und die `aws:SourceAccount` globalen Bedingungsschlüssel in Ihrer ressourcenbasierten Amazon MQ MQ-Richtlinie zu verwenden, um den CloudWatch Log-Zugriff auf einen oder mehrere angegebene Broker zu beschränken.

> **Note**
>
> Wenn Sie beide globalen Bedingungskontextschlüssel verwenden, müssen der `aws:SourceAccount`-Wert und das Konto im `aws:SourceArn`-Wert dieselbe Konto-ID verwenden, wenn sie in derselben Richtlinienanweisung verwendet werden.

Das folgende Beispiel zeigt eine ressourcenbasierte Richtlinie, die den CloudWatch Logs-Zugriff auf einen einzelnen Amazon MQ-Broker beschränkt.

```
{
                        "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-3i45j6k178l9"
                                }
                            }
                            }
                        ]
                        }
```

Sie können Ihre ressourcenbasierte Richtlinie auch so konfigurieren, dass der Zugriff auf CloudWatch Protokolle auf alle Broker in einem Konto beschränkt wird, wie im Folgenden dargestellt.

```
{
                            "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"
                                }
                                }
                            }
                            ]
                        }
```

Weitere Informationen über das Sicherheitsproblem des verwirrten Stellvertreters finden Sie unter Das Problem des verwirrten Stellvertreters im Benutzerhandbuch.

Fehlerbehebung bei der Konfiguration von CloudWatch Protokollen mit Amazon MQ

In einigen Fällen verhalten sich CloudWatch Logs möglicherweise nicht immer wie erwartet. In diesem Abschnitt erhalten Sie einen Überblick über häufige Probleme und deren Lösungen.

Protokollgruppen erscheinen nicht in CloudWatch

Fügen Sie die `CreateLogGroup`-Berechtigung Ihrem Amazon MQ-Benutzer hinzu, und starten Sie den Broker neu. Dies ermöglicht Amazon MQ, die Protokollgruppe zu erstellen.

Protokollstreams werden nicht in CloudWatch Protokollgruppen angezeigt

Konfigurieren einer ressourcenbasierten Richtlinie für Amazon MQ. Dies ermöglicht es Ihrem Broker, seine Protokolle zu veröffentlichen.

Kontingente in Amazon MQ

In diesem Thema werden die Beschränkungen in Amazon MQ aufgeführt. Viele der folgenden Grenzwerte können für bestimmte AWS Konten geändert werden. Weitere Informationen zur Beantragung einer Erhöhung eines Limits finden Sie unter AWS Service-Kontingente in der Allgemeine Amazon Web Services-Referenz. Aktualisierte Limits sind auch nach Anwendung der Limit-Erhöhung nicht sichtbar. Weitere Informationen zur Anzeige der aktuellen Verbindungslimits bei Amazon CloudWatch finden Sie unter Überwachung von Amazon MQ-Brokern mithilfe von Amazon CloudWatch.

Themen

Broker

In der folgenden Tabelle werden die Kontingente im Zusammenhang mit Amazon MQ-Brokern aufgeführt.

Limit	Beschreibung
Broker-Name	• Muss in Ihrem AWS Konto eindeutig sein. • Er muss 1–50 Zeichen umfassen. • Darf nur Zeichen enthalten, die im ASCIIdruckbaren Zeichensatz angegeben sind. • Er darf nur alphanumerische Zeichen, Bindestriche, Punkte, Unterstriche und Tilden (- . _ ~) enthalten.

Limit	Beschreibung
Anzahl der Broker, pro Region	50
Wire-Level-Verbindungen pro Protokoll für kleineren Broker	**Important** Gilt nicht für RabbitMQ-Broker. 300 für `mq.*.micro` Instance-Typ-Broker.
Wire-Level-Verbindungen pro Protokoll für größeren Broker	**Important** Gilt nicht für RabbitMQ-Broker. 2.000 für `mq.*.*large` Instance-Typ-Broker.
Sicherheitsgruppen pro Broker	5
ActiveMQ-Ziele (Warteschlangen und Themen) werden überwacht in CloudWatch	CloudWatch überwacht nur die ersten 1000 Ziele.
RabbitMQ-Ziele (Warteschlangen) werden überwacht in CloudWatch	CloudWatch überwacht nur die ersten 500 Ziele, sortiert nach der Anzahl der Verbraucher.
Tags pro Broker	50

Konfigurationen

In der folgenden Tabelle werden die Kontingente im Zusammenhang mit Amazon MQ-Konfigurationen aufgeführt.

Important
Gilt nicht für RabbitMQ-Broker.

Limit	Beschreibung
Konfigurationsname	• Er muss 1–150 Zeichen umfassen. • Darf nur Zeichen enthalten, die im ASCIIdruckbaren Zeichensatz angegeben sind. • Er darf nur alphanumerische Zeichen, Bindestriche, Punkte, Unterstriche und Tilden (- . _ ~) enthalten.
Revisionen pro Konfiguration	300

Benutzer

In der folgenden Tabelle werden die Kontingente im Zusammenhang mit Amazon MQ ActiveMQ-Broker-Benutzern aufgeführt.

> **Important**
>
> Gilt nicht für RabbitMQ-Broker.

Limit	Beschreibung
Username	• Er muss 1–100 Zeichen umfassen. • Darf nur Zeichen enthalten, die im ASCIIdruckbaren Zeichensatz angegeben sind. • Er darf nur alphanumerische Zeichen, Bindestriche, Punkte, Unterstriche und Tilden (- . _ ~) enthalten. • Er darf keine Kommas enthalten. (,).

Limit	Beschreibung
Passwort	• Es muss 12–250 Zeichen umfassen. • Darf nur Zeichen enthalten, die im ASCIIdruckbaren Zeichensatz angegeben sind. • Es muss mindestens 4 eindeutige Zeichen enthalten. • Es darf keine Kommas enthalten. (`,`).
Benutzer pro Broker (einfache Auth)	250
Gruppen pro Benutzer (einfache Auth)	20

Datenspeicherung

In der folgenden Tabelle werden die Kontingente im Zusammenhang mit der Amazon MQ-Datenspeicherung aufgeführt.

Limit	Beschreibung
Speicherkapazität pro kleinerem Broker	20 GB für `mq.*.micro` Instance-Typ-Broker. Weitere Informationen zu Amazon MQ Instance-Typen finden Sie unter Broker instance types.
Speicherkapazität pro Broker	200 GB für `mq.*.*large` Instance-Typ-Broker. Weitere Informationen zu Amazon MQ Instance-Typen finden Sie unter Broker instance types.
Von Amazon unterstütztes Nutzungslimit für Job Scheduler pro Broker EBS	

Limit	Beschreibung
	Important Gilt nicht für RabbitMQ-Broker. 50 GB. Weitere Informationen zur Verwendung des Job Schedulers finden Sie `JobSchedulerUsage` in der Apache ActiveMQ-Dokumentation API.
Temporäre Speicherkapazität pro kleineren Broker.	**Important** Gilt nicht für RabbitMQ-Broker. 5 GB `mq.*.micro` Instance-Typ-Broker.
Temporäre Speicherkapazität pro größeren Broker.	**Important** Gilt nicht für RabbitMQ-Broker. 50 GB für `mq.*.*large` Instance-Typ-Broker.

APIDrosselung

Die folgenden Drosselungskontingente werden pro AWS Konto für alle Amazon MQs zusammengefasst, APIs um die Servicebandbreite aufrechtzuerhalten. Weitere Informationen zu Amazon MQ APIs finden Sie in der Amazon MQ MQ-Referenz REST API.

> **Important**
>
> Diese Kontingente gelten nicht für Amazon MQ for ActiveMQ oder Amazon MQ for RabbitMQ Broker Messaging. APIs Amazon MQ drosselt z. B. nicht das Senden und Empfangen von Nachrichten.

APIBurst-Limit	APIRatenbegrenzung
100	15

Fehlerbehebung für Amazon MQ

Dieser Abschnitt beschreibt häufige Probleme, die beim Verwenden von Amazon MQ-Brokern auftreten können, und was Sie tun müssen, um diese zu lösen.

Fehlerbehebung: Allgemeines Amazon MQ

Verwenden Sie die Informationen in diesem Abschnitt, um häufige Probleme zu diagnostizieren, die beim Arbeiten mit Amazon MQ-Brokern auftreten können, z. B. Probleme beim Herstellen der Verbindung mit Ihrem Broker und Neustarts von Broker.

Inhalt

Ich kann keine Verbindung zu meiner Broker-Webkonsole oder -Endpunkten herstellen.

Wenn Probleme beim Herstellen einer Verbindung mit Ihrem Broker über die Webkonsole oder Wire-Level-Endpunkte auftreten, empfehlen wir die folgenden Schritte.

1. Überprüfen Sie, ob Sie versuchen, sich hinter einer Firewall mit Ihrem Broker zu verbinden. Möglicherweise müssen Sie die Firewall so konfigurieren, dass der Zugriff auf Ihren Broker gewährt wird.
2. Prüfen Sie, ob Sie versuchen, über einen Endpunkt eine Verbindung zu Ihrem Broker herzustellen. FIPS Amazon MQ unterstützt FIPS Endpunkte nur bei der Nutzung von API Operations und nicht für Wire-Level-Verbindungen zur Broker-Instance selbst.
3. Überprüfen Sie, ob die Option Public Accessibility (öffentliche Zugänglichkeit) für Ihren Broker auf Yes (Ja) gestellt ist. Wenn diese Option auf Nein gesetzt ist, überprüfen Sie die Regeln der Netzwerk-Zugriffskontrollliste () Ihres Subnetzes. ACL Wenn Sie ein benutzerdefiniertes Netzwerk erstellt habenACLs, müssen Sie möglicherweise die ACL Netzwerkregeln ändern, um Zugriff auf

Ihren Broker zu gewähren. Weitere Informationen zu VPC Amazon-Netzwerken finden Sie unter Aktivieren des Internetzugangs im VPCAmazon-Benutzerhandbuch

4. Überprüfen Sie die Sicherheitsgruppenregeln Ihres Brokers. Stellen Sie sicher, dass Sie Verbindungen zu den folgenden Ports zulassen:

> **Note**
>
> Die folgenden Ports sind nach Modultypen gruppiert, da Amazon MQ für ActiveMQ und Amazon MQ für RabbitMQ unterschiedliche Ports für Verbindungen verwenden.

Amazon MQ für ActiveMQ

- Webkonsole — Port8162
- OpenWire — Anschluss 61617
- AMQP— Hafen 5671
- STOMP— Hafen 61614
- MQTT— Hafen 8883
- WSS— Hafen 61619

Amazon MQ

- Webkonsole und Verwaltung API — Port 443 und 15671
- AMQP— Anschluss 5671

5. Führen Sie die folgenden Netzwerkkonnektivitätstests für Ihren Broker-Engine-Typ .

> **Note**
>
> Führen Sie für Broker ohne öffentlichen Zugriff die Tests von einer EC2 Amazon-Instance innerhalb desselben Amazon VPC wie Ihr Amazon MQ-Broker aus und bewerten Sie die Antworten.

Amazon MQ for ActiveMQ

So testen Sie die Netzwerkkonnektivität Ihres Amazon MQ-Brokers

1. Öffnen Sie ein Terminal-Fenster oder eine Eingabeaufforderung.

2. Führen Sie den folgenden `nslookup` Befehl aus, um Ihren DNS Broker-Datensatz abzufragen. Füraktive/Standby-Funktion-Bereitstellungen verwenden, testen Sie sowohl die aktiven als auch die Standby-Endpunkte. Die Aktiv-/Standby-Endpunkte werden mit einem Suffix,`-1`oder`-2`der eindeutigen Broker-ID hinzugefügt. Ersetzen Sie den Endpunkt durch Ihre Informationen.

```
$ nslookup b-1234a5b6-78cd-901e-2fgh-3i45j6k178l9-1.mq.us-west-2.amazonaws.com
```

Wird der Befehl erfolgreich ausgeführt, wird Ihnen eine Ausgabe ähnlich der folgenden angezeigt:

```
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
```

Die aufgelöste IP-Adresse sollte mit den in der Amazon MQ Konsole angegebenen IP-Adressen übereinstimmen. Dies weist darauf hin, dass der Domainname auf dem DNS Server korrekt aufgelöst wird, und Sie können mit dem nächsten Schritt fortfahren.

3. Führen Sie Folgendes aus:`telnet`, um den Netzwerkpfad für Ihren Broker zu testen. Ersetzen Sie den Endpunkt durch Ihre Informationen. Ersetzen *`port`* mit der Portnummer `8162` für die Webkonsole oder anderen Anschlüssen auf Kabelebene, um bei Bedarf zusätzliche Protokolle zu testen.

> **Note**
>
> Für Acive-/Standby-Bereitstellungen erhalten Sie eine`Connect failed`Fehlermeldung, wenn Sie`telnet`mit dem Standby-Endpunkt. Dies wird erwartet, da die Standby-Instance selbst läuft, der ActiveMQ-Prozess jedoch nicht läuft und keinen Zugriff auf das EFS Amazon-Speichervolumen des Brokers hat. Führen Sie den Befehl für`-1`und`-2`-Endpunkte, um sicherzustellen, dass Sie sowohl die aktive als auch die Standby-Instance testen.

```
$ telnet b-1234a5b6-78cd-901e-2fgh-3i45j6k178l9-1.mq.us-
west-2.amazonaws.com port
```

Für die aktive Instance wird eine Ausgabe ähnlich der folgenden angezeigt.

```
Connected to b-1234a5b6-78cd-901e-2fgh-3i45j6k178l9-1.mq.us-
west-2.amazonaws.com.
Escape character is '^]'.
```

4. Führen Sie eine der folgenden Aufgaben aus.

   - Wenn der Befehl `telnet` erfolgreich ist, überprüfen Sie die Metrik `EstablishedConnectionsCount` und bestätigen Sie, dass der Broker die maximale Grenze für Wire-Limits nicht erreicht hat. Sie können auch bestätigen, ob das Limit erreicht wurde, indem Sie den Broker`General`-Protokolle. Wenn diese Metrik größer als Null ist, ist derzeit mindestens ein Client mit dem Broker verbunden. Wenn die Metrik keine Verbindungen anzeigt, führen Sie die`telnet`-Pfadtest erneut und warten Sie mindestens eine Minute, bevor Sie die Verbindung trennen, da Broker-Metriken jede Minute veröffentlicht werden.
   - Wenn das Symbol`telnet`-Befehl fehlschlägt, überprüfen Sie den Status IhrerElastic Network-Schnittstelle, und bestätigen Sie, dass der Status`in-use` ist. Erstellen Sie ein VPC Amazon-Flow-Protokoll für die Netzwerkschnittstelle jeder Instance und überprüfen Sie die generierten Flow-Protokolle. Suchen Sie nach den IP-Adressen des Brokers, wenn Sie die`telnet`und vergewissern Sie sich, dass die Verbindungspakete`ACCEPTED`, einschließlich eines Rücksendepakets. Weitere Informationen und ein Beispiel für ein Flow-Protokoll finden Sie unter Beispiele für Flow-Protokolldatensätze im Amazon VPC Developer Guide.

5. Führen Sie Folgendes aus:`curl`, um die Konnektivität zur ActiveMQ -Admin-Webkonsole zu überprüfen.

```
$ curl https://b-1234a5b6-78cd-901e-2fgh-3i45j6k178l9-1.mq.us-
west-2.amazonaws.com:8162/index.html
```

Wenn der Befehl erfolgreich ist, sollte die Ausgabe ein HTML Dokument sein, das dem folgenden ähnelt.

```
<!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

So testen Sie die Netzwerkkonnektivität Ihres Amazon MQ-Brokers

1. Öffnen Sie ein Terminal-Fenster oder eine Eingabeaufforderung.
2. Führen Sie den folgenden `nslookup` Befehl aus, um Ihren DNS Brokerdatensatz abzufragen. Ersetzen Sie den Endpunkt durch Ihre Informationen.

   ```
   $ nslookup b-1234a5b6-78cd-901e-2fgh-3i45j6k178l9-1.mq.us-west-2.amazonaws.com
   ```

   Wird der Befehl erfolgreich ausgeführt, wird Ihnen eine Ausgabe ähnlich der folgenden angezeigt:

   ```
   Non-authoritative answer:
   Server:  dns-resolver-corp-sfo-1.sfo.corp.amazon.com
   Address:  172.10.123.456

   Name:    rabbit-broker-1c23e456ca78-b9000123b4ebbab5.elb.us-
   west-2.amazonaws.com
   Addresses:  52.12.345.678
             52.23.234.56
             41.234.567.890
             54.123.45.678
   Aliases:  b-1234a5b6-78cd-901e-2fgh-3i45j6k178l9-1.mq.us-west-2.amazonaws.com
   ```

3. Führen Sie Folgendes aus:`telnet`, um den Netzwerkpfad für Ihren Broker zu testen. Ersetzen Sie den Endpunkt durch Ihre Informationen. Sie können ersetzen *`port`* mit Anschluss `443` für die Webkonsole und `5671` zum Testen der Wire-Level-VerbindungAMQP.

```
$ telnet b-1234a5b6-78cd-901e-2fgh-3i45j6k178l9-1.mq.us-
west-2.amazonaws.com port
```

Wird der Befehl erfolgreich ausgeführt, wird Ihnen eine Ausgabe ähnlich der folgenden angezeigt:

```
Connected to b-1234a5b6-78cd-901e-2fgh-3i45j6k178l9-1.mq.us-
west-2.amazonaws.com.
Escape character is '^]'.
```

> **Note**
>
> Die Telnet-Verbindung wird nach einigen Sekunden automatisch geschlossen.

4. Führen Sie eine der folgenden Aufgaben aus.

   - Wenn der `telnet`-Befehl erfolgreich ist, überprüfen Sie die `ConnectionCount`-Metrik und bestätigen Sie, dass der Broker den Wert nicht erreicht hat, der in der `max-connections`-Standardrichtlinie eingestellt ist. Sie können auch bestätigen, ob das Limit erreicht wurde, indem Sie den Broker`Connection.log`-Protokollgruppe. Wenn diese Metrik größer als Null ist, ist derzeit mindestens ein Client mit dem Broker verbunden. Wenn die Metrik keine Verbindungen anzeigt, führen Sie die`telnet`-Pfadtest erneut. Möglicherweise müssen Sie diesen Vorgang wiederholen, wenn die Verbindung geschlossen wird, bevor Ihr Broker neue Verbindungsmetriken veröffentlicht hat. CloudWatch Metriken werden alle fünf Minuten veröffentlicht.
   - Für Broker ohne öffentliche Zugänglichkeit, wenn die`telnet`-Befehl fehlschlägt, überprüfen Sie den Status IhrerElastic Network-Schnittstellen, und bestätigen Sie, dass der Status`in-use`. Erstellen Sie ein VPC Amazon-Flow-Protokoll für jede Netzwerkschnittstelle und überprüfen Sie die generierten Flow-Protokolle. Suchen Sie nach den privaten IP-Adressen des Brokers, wenn der`telnet`-Befehl aufgerufen wurde, und bestätigen Sie, dass die Verbindungspaket`ACCEPTED`, einschließlich eines Rücksendepakets. Weitere Informationen und ein Beispiel für ein Flow-Protokoll finden Sie unter Beispiele für Flow-Protokolldatensätze im Amazon VPC Developer Guide.

> **Note**
>
> Dieser Schritt gilt nicht für Amazon MQ-Broker für RabbitMQ-Broker mit öffentlicher Zugänglichkeit.

5. Führen Sie Folgendes aus:`curl`, um die Konnektivität zur RabbitMQ Admin-Webkonsole zu überprüfen.

```
$ curl https://b-1234a5b6-78cd-901e-2fgh-3i45j6k178l9-1.mq.us-west-2.amazonaws.com:443/index.html
```

Wenn der Befehl erfolgreich ist, sollte die Ausgabe ein HTML Dokument sein, das dem folgenden ähnelt.

```
<!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>
        ...
```

Mein Broker läuft und ich kann die Konnektivität mit überprüfen`telnet`, aber meine Clients können keine Verbindung herstellen und geben SSL Ausnahmen zurück.

Ihr Broker-Endpunktzertifikat wurde möglicherweise während des Wartungsfensters des Brokers aktualisiert. Amazon MQ-Brokerzertifikate werden regelmäßig rotiert, um die fortgesetzte Verfügbarkeit und Sicherheit von Brokern zu gewährleisten.

Wir empfehlen die Verwendung der Amazon-Root-Zertifizierungsstelle (CA) in Amazon Trust Services, um sich im Vertrauensspeicher Ihrer Clients zu authentifizieren. Alle Amazon-MQ-Brokerzertifikate sind mit dieser Root-CA signiert. Durch die Verwendung einer Amazon-Root-CA müssen Sie das neue Amazon-MQ-Brokerzertifikat nicht mehr jedes Mal herunterladen, wenn ein Zertifikatupdate für den Broker vorliegt.

Ich habe einen Broker erstellt, aber die Brokererstellung ist fehlgeschlagen.

Wenn sich Ihr Broker in einem`CREATION_FAILED`-Status haben, gehen Sie wie folgt vor.

- Überprüfe deine IAM Berechtigungen. Um einen Broker zu erstellen, müssen Sie entweder die AWS verwaltete IAM Richtlinie verwenden `AmazonMQFullAccess` oder über die richtigen EC2 Amazon-Berechtigungen in Ihrer benutzerdefinierten IAM Richtlinie verfügen. Weitere Informationen zu den erforderlichen EC2 Amazon-Berechtigungen, die Sie benötigen, finden Sie unter Erforderliche IAM Berechtigungen zum Erstellen eines Amazon MQ-Brokers.
- Prüfen Sie, ob sich das Subnetz, das Sie für Ihren Broker auswählen, in einer gemeinsam genutzten Amazon Virtual Private Cloud (VPC) befindet. Um einen Amazon MQ-Broker in einem gemeinsam genutzten Amazon zu erstellen`VPC`, müssen Sie ihn in dem Konto erstellen, dem der Amazon VPC gehört.

Mein Broker wurde neu gestartet und ich bin mir nicht sicher, warum.

Wird Ihr Broker automatisch neu gestartet, kann dies auf einen der folgenden Gründe zurückzuführen sein:

- Ihr Broker wurde möglicherweise aufgrund eines geplanten wöchentlichen Wartungsfensters neu gestartet. In regelmäßigen Abständen führt Amazon MQ Wartungsarbeiten an der Hardware, dem Betriebssystem oder der Engine-Software eines Nachrichtenbrokers durch. Die Dauer der Wartung variiert, kann jedoch bis zu zwei Stunden dauern, abhängig von den Vorgängen, die für den Nachrichtenbroker geplant sind. Broker können jederzeit während des zweistündigen Wartungsfensters neu starten. Weitere Informationen über Broker-Wartungsfenster finden Sie unter the section called “Planung der Wartung des Brokers”.
- Ihr Broker-Instance-Typ ist möglicherweise nicht für Ihre Anwendungs-Workload geeignet. Beispiel: Ausführen eines Produktions-Workloads auf einem `mq.t2.micro` kann dazu führen, dass dem Broker keine Ressourcen mehr zur Verfügung stehen. Eine hohe CPU Auslastung oder eine hohe Broker-Speicherauslastung können dazu führen, dass ein Broker unerwartet neu gestartet wird. Verwenden Sie die folgenden CloudWatch Messwerte für Ihren Engine-Typ, um zu sehen, wie viel Speicherplatz CPU und Speicherplatz von Ihrem Broker genutzt werden.
  - Amazon MQ for ActiveMQ — Prüfen Sie `CpuUtilization` den Prozentsatz der zugewiesenen EC2 Amazon-Recheneinheiten, die der Broker derzeit verwendet. `HeapUsage`Überprüfen Sie den Prozentsatz des JVM ActiveMQ-Speicherlimits, den der Broker derzeit verwendet.

- Amazon MQ for RabbitMQ — Prüfen Sie `SystemCpuUtilization` den Prozentsatz der zugewiesenen EC2 Amazon-Recheneinheiten, die der Broker derzeit verwendet. Prüfen Sie `RabbitMQMemUsed` das RAM verwendete Volumen in Byte und dividieren Sie durch den Prozentsatz des vom `RabbitMQMemLimit` RabbitMQ-Knoten genutzten Speichers.

Weitere Informationen zu Broker-Instance-Typen und zur Auswahl des richtigen Instance-Typs für Ihre Workload finden Sie unterBroker instance types.

Fehlerbehebung bei Amazon MQ für ActiveMQ

Verwenden Sie die Informationen in diesem Abschnitt, um häufige Probleme zu diagnostizieren und zu lösen, die beim Arbeiten mit Amazon MQ for ActiveMQ Broker auftreten können.

Inhalt

Ich kann in CloudWatch Logs keine allgemeinen Logs oder Audit-Logs für meinen Broker sehen, obwohl ich die Protokollierung aktiviert habe.

Wenn Sie in CloudWatch Logs keine Logs für Ihren Broker einsehen können, gehen Sie wie folgt vor.

1. Überprüfen Sie, ob der Benutzer, der den Broker erstellt oder neu startet, über die `logs:CreateLogGroup`-Berechtigung verfügt. Wenn Sie die `CreateLogGroup`-Berechtigung nicht zu einem Benutzer hinzufügen, bevor der Benutzer den Broker erstellt oder neu startet, wird die Protokollgruppe nicht von Amazon MQ erstellt.
2. Prüfen Sie, ob Sie eine ressourcenbasierte Richtlinie konfiguriert haben, die es Amazon MQ ermöglicht, Protokolle in Logs zu veröffentlichen. CloudWatch Damit Amazon MQ Protokolle in Ihrer Logs-Protokollgruppe veröffentlichen kann, konfigurieren Sie eine ressourcenbasierte

Richtlinie, um Amazon MQ Zugriff auf die folgenden CloudWatch Logs-Aktionen zu gewähren: CloudWatch API

- `CreateLogStream`— Erstellt einen CloudWatch Logs-Log-Stream für die angegebene Protokollgruppe.
- `PutLogEvents`— Liefert Ereignisse in den angegebenen CloudWatch Log-Log-Stream.

Weitere Informationen zur Konfiguration von Amazon MQ für ActiveMQ zur Veröffentlichung von Protokollen in Logs finden Sie unter CloudWatch Protokollierung konfigurieren.

Nach dem Neustart oder dem Wartungsfenster des Brokers kann ich keine Verbindung zu meinem Broker herstellen, obwohl der Status **RUNNING** lautet. Warum?

Es treten möglicherweise Verbindungsprobleme auf, nachdem Sie den Neustart eines Brokers eingeleitet haben, nachdem ein geplantes Wartungsfenster abgeschlossen wurde, oder in einem Fehlerereignis, bei dem die Standby-Instance aktiviert ist. In beiden Fällen werden Verbindungsprobleme nach einem Neustart des Brokers höchstwahrscheinlich durch eine ungewöhnlich große Anzahl von Nachrichten verursacht, die auf dem Amazon EFS - oder EBS Amazon-Speichervolumen Ihres Brokers gespeichert sind. Während eines Neustarts verschiebt Amazon MQ persistente Nachrichten vom Speicher in den Broker-Speicher. Um diese Diagnose zu bestätigen, können Sie die folgenden Messwerte CloudWatch für Ihren Amazon MQ for ActiveMQ-Broker überwachen:

- **`StoragePercentUsage`** – Große Prozentsätze bei oder nahe 100 % können dazu führen, dass der Broker Verbindungen ablehnt.
- **`JournalFilesForFullRecovery`** – Gibt die Anzahl der Journaldateien an, die nach einem unreinen Shutdown und Neustart erneut abgespielt werden. Ist der Wert zunehmend bzw. konstant höher als Eins, weist dies auf ungelöste Transaktionen hin, die nach dem Neustart Verbindungsprobleme verursachen können.
- **`OpenTransactionCount`** –Eine Zahl größer als Null nach einem Neustart zeigt an, dass der Broker versucht, zuvor verbrauchte Nachrichten zu speichern, was zu Verbindungsproblemen führt.

Um dieses Problem zu beheben, empfehlen wir Ihnen, Ihre XA-Transaktionen mit einem `rollback()` oder `commit()` zu lösen. Weitere Informationen sowie ein Codebeispiel zum Lösen von XA-Transaktionen mit `rollback()`, finden Sie unter Wiederherstellen von XA-Transaktionen.

Ich sehe, dass einige meiner Clients eine Verbindung zum Broker herstellen, während andere keine Verbindung herstellen können.

Wenn Ihr Broker im `RUNNING`-Status ist und einige Clients sich erfolgreich mit dem Broker verbinden können, während andere dies nicht tun können, haben Sie möglicherweise das Limit an Wire-Level-Verbindungen für den Broker erreicht. Gehen Sie wie folgt vor, um zu überprüfen, ob Sie das Wire-Level-Verbindungslimit erreicht haben:

- Überprüfen Sie die allgemeinen Broker-Protokolle für Ihren Amazon MQ for ActiveMQ-Broker unter Logs. CloudWatch Wenn das Limit erreicht wurde, sehen Sie `Reached Maximum Connections` in den Broker-Protokollen. Weitere Informationen zu CloudWatch Protokollen für Amazon MQ für ActiveMQ-Broker finden Sie unter. the section called “Grundlegendes zur Struktur der Protokollierung von Protokollen CloudWatch ”

Sobald das Limit für Wire-Level-Verbindungen erreicht ist, lehnt der Broker aktiv zusätzliche eingehende Verbindungen ab. Um dieses Problem zu lösen, empfehlen wir, den Broker-Instance-Typ zu aktualisieren. Weitere Informationen zur Auswahl des besten Instance-Typs für Ihre Workload finden Sie unter Broker instance types.

Wenn Sie bestätigt haben, dass die Anzahl Ihrer Wire-Level-Verbindungen unter dem Verbindungslimit des Brokers liegt, kann das Problem mit dem Neustart von Clients zusammenhängen. Überprüfen Sie Ihre Broker-Protokolle auf zahlreiche und häufige Einträge von `... Inactive for longer than 600000 ms - removing ...`. Der Protokolleintrag weist auf einen Neustart von Clients oder Konnektivitätsprobleme hin. Dieser Effekt tritt deutlicher zutage, wenn Clients über einen Network Load Balancer (NLB) eine Verbindung zum Broker herstellen, wobei die Clients häufig die Verbindung zum Broker trennen und erneut herstellen. Dies wird typischerweise bei containerbasierten Clients beobachtet.

Weitere Informationen finden Sie in Ihren clientseitigen Protokollen. Der Broker bereinigt inaktive TCP Verbindungen nach 600000 ms und gibt den Verbindungs-Socket frei.

Ich sehe beim Ausführen von Operationen die Ausnahme **`org.apache.jasper.JasperException: An exception occurred processing JSP page`** auf der ActiveMQ-Konsole.

Wenn Sie die einfache Authentifizierung und Konfiguration `AuthorizationPlugin` für die Warteschlangen- und Themenautorisierung verwenden, stellen Sie sicher, dass Sie das

`AuthorizationEntries` Element in Ihrer XML Konfigurationsdatei verwenden und der `activemq-webconsole` Gruppe Zugriff auf alle Warteschlangen und Themen gewähren. Dies stellt sicher, dass die ActiveMQ-Webkonsole mit dem ActiveMQ-Broker kommunizieren kann.

Das folgende Beispiel-`AuthorizationEntry` erteilt Lese- und Schreibberechtigungen für alle Warteschlangen und Themen an die `activemq-webconsole`-Gruppe.

```
<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>
```

Achten Sie auch bei der Integration Ihres Brokers daraufLDAP, der `amazonmq-console-admins` Gruppe die entsprechenden Berechtigungen zu erteilen. Weitere Informationen zur LDAP Integration finden Sie unterthe section called “Funktionsweise der LDAP-Integration”.

Fehlerbehebung: Amazon MQ für RabbitMQ

Verwenden Sie die Informationen in diesem Abschnitt, um häufige Probleme zu diagnostizieren und zu lösen, die beim Arbeiten mit Amazon MQ for RabbitMQ Broker auftreten können.

Inhalt

Ich kann keine Metriken für meine Warteschlangen oder virtuellen Hosts in CloudWatch sehen.

Wenn Sie keine Metriken für Ihre Warteschlangen oder virtuellen Hosts anzeigen können CloudWatch, überprüfen Sie, ob Ihre Warteschlangen- ASCII oder virtuellen Hostnamen Leerzeichen, Tabulatoren oder andere Nichtzeichen enthalten.

Amazon MQ kann keine Metriken für virtuelle Hosts und Warteschlangen veröffentlichen, deren Namen Leerzeichen, Tabulatoren oder andere Nichtzeichen enthalten. ASCII

Weitere Informationen zu Dimensionsnamen finden Sie unter Dimension in der CloudWatch APIAmazon-Referenz.

Wie aktiviere ich Plugins in Amazon MQ für RabbitMQ?

Amazon MQ für RabbitMQ unterstützt derzeit nur die RabbitMQ-Plugins für Verwaltung, Schaufel, Verbund und konsistenten Hash-Austausch, die standardmäßig aktiviert sind. Weitere Informationen zur Verwendung unterstützter Plugins finden Sie unter the section called “Plug-ins”.

Ich kann die VPC Amazon-Konfiguration für den Broker nicht ändern.

Amazon MQ unterstützt keine Änderung der VPC Amazon-Konfiguration, nachdem Ihr Broker erstellt wurde. Bitte beachten Sie, dass Sie einen neuen Broker mit der neuen VPC Amazon-Konfiguration erstellen und die Client-Verbindung URL mit der neuen Broker-Verbindung aktualisieren müssenURL.

Amazon MQ für RabbitMQ: Alarm über hohe Speicherauslastung

RabbitMQ löst einen hohen Speicheralarm aus, wenn die Speicherauslastung des Brokers, die anhand der CloudWatch Metrik identifiziert wird, das Speicherlimit überschreitet`RabbitMQMemUsed`, das durch identifiziert wird. `RabbitMQMemLimit` `RabbitMQMemLimit`wird von Amazon MQ festgelegt und wurde speziell unter Berücksichtigung des für jeden Host-Instance-Typ verfügbaren Speichers optimiert.

Ein Broker von Amazon MQ for RabbitMQ, der einen Alarm über hohe Speicherauslastung ausgelöst hat, blockiert alle Clients, die Nachrichten veröffentlichen. Aufgrund der hohen Speicherauslastung kann es bei Ihrem Broker auch andere Probleme geben, die die Diagnose und Auflösung des Alarms erschweren.

Broker für Einzel-Instances, die aufgrund der hohen Speicherauslastung den Start nicht abschließen können, gelangen möglicherweise in eine Neustartschleife, bei der die Interaktionen mit dem Broker begrenzt sind. In Clusterbereitstellungen kann es bei Warteschlangen zu einer pausierten Synchronisierung von Nachrichten zwischen Replikaten auf verschiedenen Knoten kommen. Pausierte Warteschlangensynchronisierungen verhindern den Verbrauch von Nachrichten aus Warteschlangen und müssen separat angesprochen werden, während der Speicheralarm aufgelöst wird.

Amazon MQ startet einen Broker nicht neu, bei dem ein Alarm zu viel Arbeitsspeicher auftritt, und gibt bei `RebootBroker`APIVorgängen eine Ausnahme zurück, solange der Broker den Alarm auslöst.

Die Informationen in diesem Abschnitt helfen Ihnen bei der Diagnose und Behebung von RabbitMQ-Alarmen über hohe Speicherauslastung, die von Ihrem Broker ausgelöst werden.

> **Note**
>
> Es kann mehrere Stunden dauern, bis der ALARM Status RABBITMQ _ MEMORY _ gelöscht wird, nachdem Sie die erforderlichen Maßnahmen ergriffen haben.

> **Note**
>
> Sie können einen Broker nicht von einem `mq.m5.`-Instance-Typ auf einen `mq.t3.micro`-Instance-Typ herunterstufen. Wenn Sie ein Downgrade durchführen möchten, müssen Sie Ihren Broker löschen und einen neuen erstellen.

Themen

Diagnostizieren eines Alarms über hohe Speicherauslastung mit der RabbitMQ-Webkonsole

Die RabbitMQ-Webkonsole kann detaillierte Informationen zur Speicherauslastung für jeden Knoten generieren und anzeigen. Sie finden diese Informationen durch das folgende Verfahren:

1. Melden Sie sich an AWS Management Console und öffnen Sie die RabbitMQ-Webkonsole Ihres Brokers.
2. Auf der RabbitMQ-Konsole wählen Sie auf der Seite Übersicht den Namen eines Knotens aus der Knoten-Liste aus.

3. Wählen Sie auf der Detailseite des Knotens die Option Details zum Speicher, um den Abschnitt zu erweitern und die Informationen zur Speicherauslastung des Knotens anzuzeigen.

Die Informationen zur Speicherauslastung, die RabbitMQ in der Webkonsole bereitstellt, können Ihnen helfen, festzustellen, welche Ressourcen möglicherweise zu viel Speicher verbrauchen und zum Alarm über hohe Speicherauslastung beitragen. Weitere Informationen zu den über die RabbitMQ-Web-Konsole verfügbaren Speicherauslastung finden Sie unter Gründe für die Speichernutzung auf der Website der RabbitMQ-Server-Dokumentation.

Diagnose eines Alarms über hohe Speicherauslastung mithilfe von Amazon-MQ-Metriken

Amazon MQ aktiviert standardmäßig Metriken für Ihren Broker. Sie können Ihre Broker-Metriken einsehen, indem Sie auf die CloudWatch Konsole zugreifen oder die verwenden. CloudWatch API Die folgenden Metriken sind beim Diagnostizieren des RabbitMQ-Alarms über hohe Speicherauslastung nützlich.

Amazon MQ-Metrik CloudWatch	Grund für eine hohe Speicherauslastung	
`MessageCount`	Nachrichten werden im Speicher gespeichert, bis sie verbraucht oder verworfen werden. Eine hohe Nachrichtenanzahl kann auf eine Überauslastung der Ressourcen hinweisen und zu einem Alarm über hohe Speicherauslastung führen.	
`QueueCount`	Warteschlangen werden im Speicher gespeichert, und eine hohe Anzahl von Warteschlangen kann zu einem Alarm über hohe Speicherauslastung führen.	

Amazon MQ-Metrik CloudWatch	Grund für eine hohe Speicherauslastung	
`ConnectionCount`	Clientverbindungen nutzen Speicher, und zu viele gleichzeitige Verbindungen können zu einem Alarm über hohe Speicherauslastung führen.	
`ChannelCount`	Ähnlich wie bei Verbindungen werden Kanäle, die mit jeder Verbindung hergestellt werden, auch im Knotenspeicher gespeichert, und eine hohe Anzahl von Kanälen kann zu einem Alarm über hohe Speicherauslastung führen.	
`ConsumerCount`	Für jeden Verbraucher, der mit dem Broker verbunden ist, wird eine bestimmte Anzahl von Nachrichten aus dem Speicher in den Arbeitsspeicher geladen, bevor sie an den Verbraucher übermittelt werden. Eine große Anzahl von Verbraucherverbindungen kann zu einer hohen Speicherauslastung führen und zu einem hohen Alarm über hohe Speicherauslastung führen.	

Amazon MQ-Metrik CloudWatch	Grund für eine hohe Speicherauslastung	
`PublishRate`	Beim Veröffentlichen von Nachrichten wird der Arbeitsspeicher des Brokers genutzt. Wenn die Rate, mit der Nachrichten an den Broker veröffentlicht werden, zu hoch ist und die Rate, mit der der Broker Nachrichten an Verbraucher übermittelt, erheblich übersteigt, kann der Broker Alarm über hohe Speicherauslastung auslösen.	

Umgang mit dem Alarm über hohe Speicherauslastung

Für jeden Mitwirkenden, den Sie identifizieren, empfehlen wir die folgenden Aktionen, um den Alarm über hohe Speicherauslastung des Brokers zu mildern und aufzulösen.

Grund für eine hohe Speicherauslastung	Amazon MQ-Empfehlung	
Die Anzahl der Nachrichten in der Warteschlange ist zu hoch.	Führen Sie eine der folgenden Aktionen aus: • Verbrauchen Sie Nachrichten, die in den Warteschlangen veröffentlicht wurden. • Löschen Sie Nachrichten aus den Warteschlangen. • Löschen Sie die Warteschlangen aus Ihrem Broker.	

Grund für eine hohe Speicherauslastung	Amazon MQ-Empfehlung	
Die Anzahl der auf dem Broker konfigurierten Warteschlangen ist zu hoch.	Reduzieren Sie die Anzahl der Warteschlangen.	
Die Anzahl der auf dem Broker hergestellten Verbindungen ist zu hoch.	Reduzieren Sie die Anzahl der Verbindungen. Weitere Informationen finden Sie unter the section called “Reduzierung der Anzahl der Verbindungen und Kanäle”.	
Die Anzahl der auf dem Broker festgelegten Kanäle ist zu hoch.	Reduzieren Sie die Anzahl der Kanäle. Weitere Informationen finden Sie unter the section called “Reduzierung der Anzahl der Verbindungen und Kanäle”.	
Die Anzahl der Verbraucher, die mit dem Broker verbunden sind, ist zu hoch.	Reduzieren Sie die Gesamtzahl der Verbraucher, die mit dem Broker verbunden sind.	
Die Veröffentlichungsrate von Nachrichten ist zu hoch.	Reduzieren Sie die Rate, mit der Nachrichten an den Broker veröffentlicht werden.	
Die Rate der Clientverbindungsversuche ist zu hoch.	Reduzieren Sie die Häufigkeit, mit der Clients versuchen, sich mit dem Broker zu verbinden, um Nachrichten zu veröffentlichen oder zu konsumieren, oder konfigurieren Sie den Broker.	

Reduzierung der Anzahl der Verbindungen und Kanäle

Verbindungen zu Ihrem Broker von Amazon MQ for RabbitMQ können entweder von Ihren Clientanwendungen oder durch manuelles Schließen über die RabbitMQ-Webkonsole geschlossen werden. Um eine Verbindung mit der RabbitMQ-Webkonsole zu schließen, gehen Sie wie folgt vor.

1. Melden Sie sich bei der RabbitMQ-Webkonsole Ihres Brokers an AWS Management Console und öffnen Sie sie.
2. Wählen Sie auf der RabbitMQ-Konsole die Registerkarte Verbindungen.
3. Wählen Sie auf der Seite Verbindungen unter Alle Verbindungen den Namen der Verbindung aus, die Sie aus der Liste schließen möchten.
4. Wählen Sie auf der Seite der Verbindungsdetails die Option Diese Verbindung schließen aus, um den Abschnitt zu erweitern, wählen Sie dann Schließen erzwingen aus. Optional können Sie den Standardtext für den Grund durch eine eigene Beschreibung ersetzen. Amazon MQ for RabbitMQ gibt den von Ihnen angegebenen Grund an den Client zurück, wenn Sie die Verbindung schließen.
5. Klicken Sie im Dialogfeld auf OK, um die Verbindung zu bestätigen und zu schließen.

Wenn Sie eine Verbindung schließen, werden alle Kanäle, die mit einer geschlossenen Verbindung verbunden sind, ebenfalls geschlossen.

> **Note**
>
> Ihre Clientanwendungen sind möglicherweise so konfiguriert, dass sie Verbindungen zum Broker automatisch wiederherstellen, nachdem sie geschlossen wurden. In diesem Fall reicht das Schließen von Verbindungen von der Broker-Webkonsole nicht aus, um die Verbindungs- oder Kanalanzahl zu reduzieren.

Bei Brokern ohne öffentlichen Zugriff können Sie Verbindungen vorübergehend blockieren, indem Sie eingehenden Datenverkehr auf dem entsprechenden Nachrichtenprotokoll-Port, z. B. dem Port für Verbindungen, verweigern. `5671` AMQP Sie können den Port in der Sicherheitsgruppe blockieren, die Sie Amazon MQ beim Erstellen des Brokers zur Verfügung gestellt haben. Weitere Informationen zum Ändern Ihrer Sicherheitsgruppe finden Sie unter Regeln zu einer Sicherheitsgruppe hinzufügen im VPCAmazon-Benutzerhandbuch.

Umgang mit pausierten Warteschlangensynchronisierungen in Clusterbereitstellungen

Während Sie sich um die Alarme über hohe Speicherauslastung von RabbitMQ kümmern, stellen Sie möglicherweise fest, dass Nachrichten in einer oder mehreren Warteschlangen nicht verbraucht werden können. Diese Warteschlangen synchronisieren möglicherweise Nachrichten zwischen Knoten, in denen die jeweiligen Warteschlangen für die Veröffentlichung und den Verbrauch nicht verfügbar sind. Warteschlangensynchronisierungen können aufgrund des Alarms über hohe Speicherauslastung pausiert werden und sogar zum Arbeitsspeicheralarm beitragen.

Informationen zum Stoppen und erneuten Versuchen der Synchronisierung von pausierten Warteschlangen finden Sie unter the section called “Beheben der angehaltenen Warteschlangen-Synchronisierung”.

Umgang mit Neustartschleifen in Einzel-Instance-Brokern

Ein Einzel-Instance-Broker von Amazon MQ for RabbitMQ, der einen Alarm über hohe Speicherauslastung auslöst, läuft Gefahr, dass er nicht verfügbar ist, wenn er neu gestartet wird und nicht genügend Arbeitsspeicher zum Starten hat. Dies kann dazu führen, dass RabbitMQ in eine Neustartschleife gelangt und weitere Interaktionen mit dem Broker solange verhindert, bis das Problem behoben ist. Wenn sich Ihr Broker in einer Neustartschleife befindet, können Sie die von Amazon MQ empfohlenen Aktionen, die zuvor in diesem Abschnitt beschrieben wurden, nicht anwenden, um den Alarm über hohe Speicherauslastung zu beheben.

Um Ihren Broker wiederherzustellen, empfehlen wir, auf einen größeren Instance-Typ mit mehr Arbeitsspeicher zu aktualisieren. Im Gegensatz zu Clusterbereitstellungen können Sie einen Einzel-Instance-Broker aktualisieren, während ein Alarm über hohe Speicherauslastung auftritt, da während eines Neustarts keine Warteschlangensynchronisierungen zwischen Knoten durchgeführt werden müssen.

Verhindern von Alarmen über hohe Speicherauslastung

Für jeden von Ihnen identifizierten Faktor empfehlen wir die folgenden Maßnahmen zur Verhinderung und Verringerung des Auftretens von RabbitMQ-Alarmen über hohe Speicherauslastung.

Grund für eine hohe Speicherauslastung	Amazon-MQ-Empfehlung	
Die Anzahl der Nachrichten in der Warteschlange ist zu hoch.	Gehen Sie wie folgt vor: • Aktivieren Sie Verzögerungswarteschlangen. • Legen Sie den Grenzwert für die Warteschlangentiefe fest oder verringern Sie ihn.	
Die Anzahl der auf dem Broker konfigurierten Warteschlangen ist zu hoch.	Legen Sie den Grenzwert für die Warteschlangenanzahl fest oder verringern Sie ihn.	
Die Anzahl der auf dem Broker hergestellten Verbindungen ist zu hoch.	Legen Sie den Grenzwert für die Verbindungsanzahl fest oder verringern Sie ihn.	
Die Anzahl der auf dem Broker festgelegten Kanäle ist zu hoch.	Legen Sie eine maximale Anzahl von Kanälen pro Verbindung für Clientanwendungen fest.	
Die Anzahl der Verbraucher, die mit dem Broker verbunden sind, ist zu hoch.	Legen Sie einen geringen Vorabrufgrenzwert für Verbraucher fest.	
Die Rate der Clientverbindungsversuche ist zu hoch.	Verwenden Sie langlebigere Verbindungen, um die Anzahl und Häufigkeit von Verbindungsversuchen zu reduzieren.	

Nachdem der Arbeitsspeicheralarm Ihres Brokers behoben wurde, können Sie Ihren Host-Instance-Typ auf eine Instance mit zusätzlichen Ressourcen aktualisieren. Informationen zur Aktualisierung des Instance-Typs Ihres Brokers finden Sie `UpdateBrokerInput` in der Amazon MQ REST API MQ-Referenz.

Eine vollständige Liste der Broker-Instance-Typen finden Sie unter the section called “Instance-Typen von Amazon MQ für RabbitMQ”.

Amazon MQ für RabbitMQ: Ungültiger Schlüssel AWS Key Management Service

Amazon MQ for RabbitMQ gibt den Code INVALID _ KMS _ KEY Critical Action Required aus, wenn ein mit einem Kunden verwalteter AWS KMS key(CMK) erstellter Broker feststellt, dass der Schlüssel AWS Key Management Service (KMS) deaktiviert ist. Ein RabbitMQ-Broker mit einem überprüft CMK regelmäßig, ob der KMS Schlüssel aktiviert ist und der Broker über alle erforderlichen Zuschüsse verfügt. Wenn RabbitMQ nicht überprüfen kann, ob der Schlüssel aktiviert ist, wird der Broker unter Quarantäne gestellt und RabbitMQ gibt _ _ zurück. INVALID KMS KEY

Ohne einen aktiven KMS Schlüssel verfügt der Broker nicht über grundlegende Berechtigungen für vom Kunden verwaltete Schlüssel. KMS Der Broker kann mit Ihrem Schlüssel solange keine kryptografischen Operationen ausführen, bis Sie Ihren Schlüssel erneut aktivieren und der Broker neu gestartet wird. Ein RabbitMQ-Broker mit einem deaktivierten KMS Schlüssel wird unter Quarantäne gestellt, um eine Verschlechterung zu verhindern. Nachdem RabbitMQ feststellt, dass der KMS Schlüssel wieder aktiv ist, wird Ihr Broker aus der Quarantäne entfernt. Amazon MQ startet einen Broker mit einem deaktivierten KMS Schlüssel nicht neu und gibt eine Ausnahme für `RebootBroker` API Operationen zurück, solange der Broker weiterhin über einen ungültigen KMS Schlüssel verfügt.

Diagnose und Adressierung von _ _ INVALID KMS KEY

Um den Code für die KEY Aktion INVALID _ KMS _ zu diagnostizieren und zu beheben, müssen Sie die AWS Befehlszeilenschnittstelle (CLI) und die AWS Key Management Service Konsole verwenden.

Um Ihren Schlüssel erneut zu KMS aktivieren

1. Rufen Sie die `DescribeBroker` Methode auf, um den `kmsKeyId` für Ihren CMK Broker abzurufen.
2. Melden Sie sich bei der AWS Key Management Service Konsole an.
3. Suchen Sie auf der Seite „Vom Kunden verwaltete Schlüssel“ nach der KMS Schlüssel-ID des problematischen Brokers und überprüfen Sie, ob der Status Aktiviert lautet.
4. Wenn Ihr KMS Schlüssel deaktiviert wurde, aktivieren Sie ihn erneut, indem Sie „Schlüsselaktionen“ und anschließend „Aktivieren“ wählen. Nachdem Ihr Schlüssel erneut aktiviert wurde, müssen Sie warten, bis RabbitMQ die Quarantäne des Brokers beendet.

Um zu überprüfen, ob die erforderlichen Zuschüsse weiterhin mit dem KMS Schlüssel des Brokers verknüpft sind, rufen Sie die `ListGrant` ListGrant Methode auf, um zu überprüfen, `mq_rabbit_grant` ob `mq_grant` sie vorhanden sind. Wenn der KMS Grant oder der Schlüssel gelöscht wurde, müssen Sie den Broker löschen und einen neuen Broker mit allen erforderlichen Zuschüssen erstellen. Schritte zum Löschen eines Brokers finden Sie unter Löschen eines Brokers.

Löschen oder deaktivieren Sie einen KMS Schlüssel oder CMK Grant nicht manuell, um den Code INVALID KMS _ _ KEY Critical Action Required zu verhindern. Wenn Sie den Schlüssel löschen möchten, löschen Sie zuerst den Broker.

Amazon MQ for ActiveMQ: Elastic-Network-Schnittstellenalarm wurde gelöscht

Amazon MQ for ActiveMQ löst einen BROKER _ ENI _ DELETED Alarm aus, wenn Sie das Elastic Network Interface () eines Brokers löschen. ENI Wenn Sie zum ersten Mal einen Amazon MQ-Broker erstellen, stellt Amazon MQ eine elastic network interface in der Virtual Private Cloud (VPC) unter Ihrem Konto bereit und benötigt daher eine Reihe von EC2 Berechtigungen.

Sie dürfen diese Netzwerkschnittstelle nicht ändern oder löschen. Das Ändern oder Löschen der Netzwerkschnittstelle kann zu einem dauerhaften Verlust der Verbindung zwischen Ihnen VPC und Ihrem Broker führen. Wenn Sie die Netzwerkschnittstelle löschen möchten, müssen Sie zuerst den Broker löschen.

Amazon MQ for ActiveMQ: Alarm -wegen zu geringem Arbeitsspeicher für Broker

Amazon MQ for ActiveMQ löst einen BROKER OOM _-Alarm aus, wenn der Broker aufgrund unzureichender Speicherkapazität eine Neustartschleife durchläuft. Wenn sich ein Broker in einer Neustartschleife befindet, die auch als Unzustellbarkeitsschleife bezeichnet wird, leitet der Broker innerhalb eines kurzen Zeitfensters wiederholte Wiederherstellungsversuche ein. Broker, die aufgrund hoher Speicherauslastung den Start nicht abschließen können, gelangen möglicherweise in eine Neustartschleife, bei der die Interaktionen mit dem Broker begrenzt sind.

Amazon MQ aktiviert standardmäßig Metriken für Ihren Broker. Sie können Ihre Broker-Metriken einsehen, indem Sie auf die CloudWatch Amazon-Konsole zugreifen oder die verwenden CloudWatch API. Die folgenden Metriken sind bei der Diagnose des ActiveMQ BROKER _ -Alarms nützlich: OOM

Amazon MQ-Metrik CloudWatch	Grund für eine hohe Speicherauslastung	
`TotalMessageCount`	Nachrichten werden im Speicher gespeichert, bis sie verbraucht oder verworfen werden. Eine hohe Nachrichtenanzahl kann auf eine Überauslastung der Ressourcen hinweisen und zu einem Alarm über hohe Speicherauslastung führen.	
`HeapUsage`	Der Prozentsatz des JVM ActiveMQ-Speicherlimits, den der Broker derzeit verwendet. Ein höherer Prozentsatz weist darauf hin, dass der Broker erhebliche Ressourcen verbraucht und kann zu einem OOM Alarm führen.	
`ConnectionCount`	Clientverbindungen nutzen Speicher und zu viele gleichzeitige Verbindungen können zu einem Alarm über hohe Speicherauslastung führen.	
`CpuUtilization`	Der Prozentsatz der zugewiesenen EC2 Recheneinheiten, die der Broker derzeit verwendet.	
`TotalConsumerCount`	Für jeden Verbraucher, der mit dem Broker verbunden ist, wird eine bestimmte Anzahl	

Amazon MQ-Metrik CloudWatch	Grund für eine hohe Speicherauslastung	
	von Nachrichten aus dem Speicher in den Arbeitsspeicher geladen, bevor sie an den Verbraucher übermittelt werden. Eine große Anzahl von Verbraucherverbindungen kann einen hohen Speicherverbrauch verursachen und zu einem Alarm über hohe Speicherauslastung führen.	

Um Neustartschleifen und den BROKER OOM _-Alarm zu vermeiden, stellen Sie sicher, dass Nachrichten schnell verarbeitet werden. Dies ist möglich, indem Sie den effektivsten Broker-Instance-Typ auswählen und auch Ihre Warteschlange für unzustellbare Nachrichten bereinigen, um unzustellbare oder abgelaufene Nachrichten zu verwerfen. Weitere Informationen zur Sicherstellung einer effektiven Leistung finden Sie bei Bewährte Methoden für Amazon MQ for ActiveMQ.

Amazon MQ für RabbitMQ: Festplattenlimit-Alarm

Der Datenträgerlimit-Alarm ist ein Hinweis darauf, dass das von einem RabbitMQ-Knoten verwendete Festplattenvolumen aufgrund einer hohen Anzahl von Nachrichten, die beim Hinzufügen neuer Nachrichten nicht verbraucht wurden, gesunken ist. RabbitMQ löst einen Festplattenlimit-Alarm aus, wenn der freie Festplattenspeicher des Brokers, der anhand der CloudWatch Amazon-Metrik identifiziert wurde`RabbitMQDiskFree`, das von identifizierte Festplattenlimit erreicht. `RabbitMQDiskFreeLimit` `RabbitMQDiskFreeLimit`wird von Amazon MQ festgelegt und unter Berücksichtigung des für jeden Broker-Instance-Typ verfügbaren Festplattenspeichers definiert.

Ein Broker von Amazon MQ für RabbitMQ, der einen Festplattenlimit-Alarm ausgelöst hat, steht für die Veröffentlichung neuer Nachrichten nicht mehr zur Verfügung. Wenn RabbitMQ in einem Cluster ausgeführt wird, gilt der Festplattenalarm clusterweit. Wenn ein Knoten das Limit unterschreitet, werden eingehende Nachrichten von allen anderen Knoten blockiert. Aufgrund der mangelnden Festplattenspeichers können bei Ihrem Broker auch andere Probleme auftreten, die die Diagnose und Auflösung des Alarms erschweren.

Amazon MQ startet einen Broker, bei dem ein Festplattenalarm auftritt, nicht neu und gibt für `RebootBroker` API Operationen eine Ausnahme zurück, solange der Broker weiterhin den Alarm auslöst.

> **Note**
>
> Sie können einen Broker nicht von einem `mq.m5`-Instance-Typ auf einen `mq.t3.micro`-Instance-Typ herunterstufen. Wenn Sie ein Downgrade durchführen möchten, müssen Sie Ihren Broker löschen und einen neuen erstellen.

Diagnose und Behebung eines Festplattenlimit-Alarms

Amazon MQ aktiviert standardmäßig Metriken für Ihren Broker. Sie können Ihre Broker-Metriken einsehen, indem Sie auf die CloudWatch Amazon-Konsole zugreifen oder die verwenden CloudWatch API. `MessageCount`ist eine nützliche Metrik bei der Diagnose des RabbitMQ-Alarms zum Festplattenlimit. Nachrichten werden im Speicher gespeichert, bis sie verwendet oder verworfen werden. Eine hohe Nachrichtenanzahl weist auf eine Überauslastung des Festplattenspeichers hin und kann zu einem Festplattenalarm führen.

Verwenden Sie die Amazon MQ Managementkonsole, damit Sie den Festplattenlimit-Alarm diagnostizieren können, um:

- Verwenden Sie Nachrichten, die in den Warteschlangen veröffentlicht wurden.
- Löschen Sie Nachrichten aus den Warteschlangen.
- Löschen Sie die Warteschlangen aus Ihrem Broker.

> **Note**
>
> Es kann mehrere Stunden dauern, bis der ALARM Status RABBITMQ _ DISK _ gelöscht wird, nachdem Sie die erforderlichen Maßnahmen ergriffen haben.

Wenn Sie verhindern möchten, dass der Festplattenlimit-Alarm erneut auftritt, können Sie Ihren Host-Instance-Typ auf eine Instance mit zusätzlichen Ressourcen aktualisieren. Informationen zur Aktualisierung des Instance-Typs Ihres Brokers finden Sie `UpdateBrokerInput` in der Amazon MQ REST API MQ-Referenz.

Amazon MQ für RabbitMQ: Alarm für Quorum-Warteschlangen

Quorum-Warteschlangen werden nur auf Amazon MQ für RabbitMQ Versionen 3.13 und höher unterstützt. Amazon MQ for RabbitMQ löst den Code für kritische erforderliche Aktionen aus, `RABBITMQ_QUORUM_QUEUES_NOT_SUPPORTED_ON_CURRENT_VERSION` wenn Sie versuchen, Quorum-Warteschlangen auf einer einzelnen Instance oder einem Cluster-Broker mit Version 3.12 und niedriger zu erstellen.

Um den `RABBITMQ_QUORUM_QUEUES_NOT_SUPPORTED_ON_CURRENT_VERSION` Alarm zu diagnostizieren und zu beheben, können Sie Ihre Liste der Quorum-Warteschlangen im RabbitMQ-Management-Dashboard einsehen:

- Wenn Sie keine Nachrichten behalten müssen, können Sie die Quorum-Warteschlangen löschen, Ihren Broker auf Version 3.13 oder höher aktualisieren und die Quorum-Warteschlangen nach dem Upgrade des Brokers neu erstellen.
- Wenn Sie Nachrichten behalten müssen, müssen Sie auf Version 3.13 und höher einen neuen Broker und anschließend Quorumwarteschlangen auf dem neuen Broker erstellen. Nachdem Sie den neuen Broker und die Quorum-Warteschlangen erstellt haben, können Sie mithilfe des Shovel- oder Federation-Plug-ins Nachrichten vom alten Broker zum neuen Broker migrieren. Löschen Sie anschließend den alten Broker.

Um dies zu verhindern`RABBITMQ_QUORUM_QUEUES_NOT_SUPPORTED_ON_CURRENT_VERSION`, aktualisieren Sie Ihren Broker auf Version 3.13 oder höher, bevor Sie Quorum-Warteschlangen auf diesem Broker erstellen.

Zugehörige Ressourcen

Amazon MQ-Ressourcen

In der folgenden Tabelle werden nützliche Ressourcen für die Arbeit mit Amazon MQ aufgeführt.

Ressource	Beschreibung
Amazon MQ REST-API-Referenz	Beschreibungen der REST-Ressourcen, Beispielanfragen, HTTP-Methoden, Schemata, Parameter und die Fehler, die der Service ausgibt.
Amazon MQ in derAWS CLI-Befehlsreferenz	Beschreibungen der AWS CLI-Befehle, die Sie für die Arbeit mit Message Brokern verwenden können.
Amazon MQ im AWS CloudFormationBenutzerhandbuch	Mit der `AWS::AmazonMQ::Broker`-Ressource können Sie Amazon MQ-Broker erstellen, Konfigurationsänderungen hinzufügen oder Benutzer für den angegebenen Broker ändern, Informationen über den angegebenen Broker zurückgeben und den angegebenen Broker löschen. Mit der `AWS::AmazonMQ::Configuration`-Ressource können Sie Amazon MQ-Konfigurationen erstellen, Konfigurationsänderungen hinzufügen oder Benutzer ändern und Informationen über die angegebene Konfiguration zurückgeben.
Regionen und Endpunkte	Informationen zu Amazon MQ-Regionen und -Endpunkten
Produktseite	Hauptwebsite für Informationen zu Amazon MQ.

Ressource	Beschreibung
Diskussionsforum	Ein auf der Community basierendes Forum, das für Entwickler eingerichtet wurde, um technische Fragen zu Amazon MQ zu klären
AWS Informationen zu Premium Support	Die Hauptwebsite mit Informationen zu AWS Premium Support ist ein persönlicher und reaktionsschneller Supportkanal. Er bietet Ihnen Hilfe beim Konfigurieren und Verwenden von Anwendungen auf AWS Infrastructure Services.

Amazon MQ für ActiveMQ-Ressourcen

In der folgenden Tabelle werden nützliche Ressourcen für die Arbeit mit Apache ActiveMQ aufgeführt.

Ressource	Beschreibung
Apache ActiveMQ – Handbuch Erste Schritte	Die offizielle Dokumentation für Apache ActiveMQ.
ActiveMQ in Aktion	Ein Handbuch für Apache ActiveMQ, das den Aufbau von JMS-Nachrichten, Verbindungselementen, Mitteilungspersistenz, Authentifizierung und Autorisierung abdeckt.
Cross-Language-Clients	Eine Liste der Programmiersprachen und der entsprechenden Apache ActiveMQ-Bibliotheken. Siehe auch ActiveMQ-Client und QpidJMS-Client.

Amazon MQ für RabbitMQ-Ressourcen

In der folgenden Tabelle werden nützliche Ressourcen für die Arbeit mit RabbitMQ aufgeführt.

Ressource	Beschreibung
Das RabbitMQ Handbuch „Erste Schritte“	Die offizielle Dokumentation für RabbitMQ.
RabbitMQ Client-Bibliotheken und Entwickler-Tools	Ein Leitfaden zu den offiziell unterstützten Client-Bibliotheken und Devloper-Tools für die Arbeit mit RabbitMQ unter Verwendung einer Vielzahl von Programmiersprachen und Plattformen.
Bewährte Methoden für RabbitMQ	Bewährte Methoden und Empfehlungen für die Arbeit mit RabbitMQ.

Versionshinweise zu Amazon MQ

Die folgende Tabelle listet neu eingeführte und verbesserte Amazon MQ-Funktionen auf.

Datum	Aktualisierung der Dokumentation
25. Juli 2024	Amazon MQ unterstützt jetzt ActiveMQ 5.18, eine neue Engine-Nebenversion. Weitere Informationen finden Sie hier: • ActiveMQ 5.18-Release-Seite • Verwalten von Amazon MQ für ActiveMQ Engine-Versionen • Aktualisieren einer Amazon MQ-Broker-Engine-Version • Verwenden von XML Spring-Konfigurationsdateien
22. Juli 2024	Amazon MQ unterstützt jetzt Quorum-Warteschlangen nur für Broker, die Version 3.13 und höher verwenden. Quorum-Warteschlangen sind replizierte FIFO Warteschlangenarten, die den Raft-Konsensusalgorithmus verwenden , um die Datenkonsistenz aufrechtzuerhalten. Quorumwarteschlangen ermöglichen die Bearbeitung unberechtigter Nachrichten, was Ihnen bei der Verwaltung unverarbeiteter Nachrichten helfen kann. Informationen zu den ersten Schritten mit Quorumwarteschlangen finden Sie unter. Quorum-Warteschlangen für RabbitMQ auf Amazon MQ
2. Juli 2024	Amazon MQ for RabbitMQ unterstützt jetzt RabbitMQ 3.13, eine Nebenvers ion. Für alle Broker, die Engine-Version 3.13 und höher verwenden, verwaltet Amazon MQ während des Wartungsfensters Upgrades auf die neueste unterstützte Patch-Version. Weitere Informationen finden Sie unter Aktualisi eren einer Amazon MQ-Broker-Engine-Version. Größenrichtlinien für Amazon MQ für RabbitMQwurden aktualisiert und enthalten nun neue Grenzwerte für Warteschlangen, Verbraucher pro Kanal und Schaufeln für Makler, die Engine-Version 3.13 verwenden. Weitere Informationen zu den Fixes und Funktionen in dieser Version finden Sie in den RabbitMQ 3.13-Versionshinweisen im RabbitMQ-Server-Re pository. GitHub

Datum	Aktualisierung der Dokumentation
	Weitere Informationen zu den unterstützten Amazon MQ-für-RabbitMQ-Versionen und Broker-Upgrades finden Sie unter Verwalten von Amazon-MQ-für-RabbitMQ-Engine-Versionen.
10. Juni 2024	Amazon MQ ist jetzt in der Region Kanada West (Calgary) verfügbar. Informationen über die verfügbaren Regionen finden Sie unter AWS -Regionen und -Endpunkte im Allgemeinen AWS -Referenzleitfaden.
10. Mai 2024	Der Support-Kalender für die Amazon MQ MQ-Version gibt an, wann der Support für eine Broker-Engine-Version endet. Wenn der Support für eine Engine-Version endet, aktualisiert Amazon MQ alle Broker der Version automatisch auf die nächste unterstützte Nebenversion. Amazon MQ informiert Sie mindestens 90 Tage im Voraus, bevor der Support für eine Engine-Version endet. Den Kalender für den Versionssupport und das Ende des Supports finden Sie im Folgenden: • Verwalten von Amazon MQ für ActiveMQ Engine-Versionen • Verwalten von Amazon-MQ-für-RabbitMQ-Engine-Versionen Sie können auch automatische Upgrades für Nebenversionen aktivieren, damit Ihr Broker während eines Wartungsfensters auf die nächste Patch-Version aktualisiert. Weitere Informationen finden Sie unter Aktualisieren einer Amazon MQ-Broker-Engine-Version

Datum	Aktualisierung der Dokumentation
9. Mai 2024	Amazon MQ for RabbitMQ unterstützt jetzt RabbitMQ 3.12, eine Nebenversion. Alle Broker auf Version 3.12.13 und höher verwenden Classic Queues Version 2 (CQv2), und alle Warteschlangen auf Version 3.12.13 und höher verhalten sich wie faule Warteschlangen. Wir empfehlen Brokern mit Versionen vor 3.12.13 Enable CQv2 und Lazy Queues oder ein Upgrade auf die neueste Version von Amazon MQ for RabbitMQ. Weitere Informationen zu den Korrekturen und Features in diesem Release finden Sie hier: • RabbitMQ 3.12-Versionshinweise zum RabbitMQ-Server-Repository. GitHub • Aktivieren von Classic Queue v2 für Ihren RabbitMQ-Broker • Lazy-Warteschlangen aktivieren Weitere Informationen zu den unterstützten Amazon MQ-für-RabbitMQ-Versionen und Broker-Upgrades finden Sie unter Verwalten von Amazon-MQ-für-RabbitMQ-Engine-Versionen.
4. März 2024	Amazon MQ für RabbitMQ unterstützt jetzt RabbitMQ 3.11.28. Weitere Informationen zu den Korrekturen und Features in diesem Release finden Sie hier: • RabbitMQ 3.11.28 Versionshinweise zum RabbitMQ-Server-Repository GitHub • RabbitMQ-Änderungsprotokoll Weitere Informationen zu den unterstützten Amazon MQ-für-RabbitMQ-Versionen und Broker-Upgrades finden Sie unter Verwalten von Amazon-MQ-für-RabbitMQ-Engine-Versionen.

Datum	Aktualisierung der Dokumentation
19. Januar 2024	Amazon MQ for RabbitMQ unterstützt den Benutzernamen „guest“ nicht und löscht das Standard-Gastkonto, wenn Sie einen neuen Broker erstellen. Amazon MQ löscht außerdem regelmäßig alle vom Kunden erstellten Konten mit dem Namen „Gast“.
15. Dezember 2023	Amazon MQ ist jetzt in der Region Israel (Tel Aviv) verfügbar. Informationen über die verfügbaren Regionen finden Sie unter AWS -Regionen und -Endpunkte im Allgemeinen AWS -Referenzleitfaden.
11. Dezember 202	Amazon MQ für RabbitMQ unterstützt jetzt RabbitMQ 3.10.25. Weitere Informationen zu den Korrekturen und Features in diesem Release finden Sie hier: • RabbitMQ 3.10.25 Versionshinweise zum RabbitMQ-Server-Repository GitHub • RabbitMQ-Änderungsprotokoll Weitere Informationen zu den unterstützten Amazon MQ-für-RabbitMQ-Versionen und Broker-Upgrades finden Sie unter Verwalten von Amazon-MQ-für-RabbitMQ-Engine-Versionen.
26. Oktober 2023	Amazon MQ hat die neuesten ActiveMQ-Nebenversionen 5.15.16, 5.16.7, 5.17.6 mit einem wichtigen Update veröffentlicht. Wir haben die älteren Nebenversionen von ActiveMQ als veraltet eingestuft und werden alle Broker auf allen Versionen von 5.15 auf 5.15.16 bzw. von 5.16 auf 5.16.7 und von 5.17 auf 5.17.6 aktualisieren. Weitere Informationen zur Aktualisierung Ihres ActiveMQ-Brokers finden Sie unter Verwalten von Amazon MQ für ActiveMQ Engine-Versionen.

Datum	Aktualisierung der Dokumentation
27. September 202	Amazon MQ für RabbitMQ unterstützt jetzt RabbitMQ 3.11.20. Weitere Informationen zu den Korrekturen und Features in diesem Release finden Sie hier: • RabbitMQ 3.11.20 Versionshinweise zum RabbitMQ-Server-Repository GitHub • RabbitMQ-Änderungsprotokoll Weitere Informationen zu den unterstützten Amazon MQ-für-RabbitMQ-Versionen und Broker-Upgrades finden Sie unter Verwalten von Amazon-MQ-für-RabbitMQ-Engine-Versionen.
27. Juli 2023	Amazon MQ für RabbitMQ unterstützt jetzt RabbitMQ 3.11.16. Weitere Informationen zu den Korrekturen und Features in diesem Release finden Sie hier: • RabbitMQ 3.11.16 Versionshinweise zum RabbitMQ-Server-Repository GitHub • RabbitMQ-Änderungsprotokoll Weitere Informationen zu den unterstützten Amazon MQ-für-RabbitMQ-Versionen und Broker-Upgrades finden Sie unter Verwalten von Amazon-MQ-für-RabbitMQ-Engine-Versionen.
27. Juli 2023	Amazon MQ für RabbitMQ unterstützt jetzt das Erstellen und Anwenden von Konfigurationen auf Ihren RabbitMQ-Broker. Weitere Informationen zum Hinzufügen von Konfigurationen zu Ihrem Broker finden Sie unter RabbitMQ Broker Configurations. Weitere Informationen über dieses Feature finden Sie unter: • Richtlinien für Betreiber • Änderungen der Richtlinien für Betreiber

Datum	Aktualisierung der Dokumentation
23. Juni 2023	Amazon MQ unterstützt jetzt ActiveMQ 5.17.3, eine neue Engine-Unterversio n. Diese Version unterstützt die neue Funktion zur regionsübergreifenden Datenreplikation (CRDR) von Amazon MQ. Weitere Informationen finden Sie hier: • Informationen zu den ersten Schritten CRDR finden Sie Regionsübergreifen de Datenreplikation für Amazon MQ für ActiveMQ im Entwicklerhandbuch. • ActiveMQ 5.17.3 – Versionsseite • Verwalten von Amazon MQ für ActiveMQ Engine-Versionen • Aktualisieren einer Amazon MQ-Broker-Engine-Version • Verwenden von XML Spring-Konfigurationsdateien
21. Juni 2023	Amazon MQ for ActiveMQ bietet jetzt eine Funktion zur regionsübergreifenden Datenreplikation (CRDR), die eine asynchrone Nachrichtenreplikation vom primären Broker in einer primären AWS Region zum Replikatbroker in einer Replikatregion ermöglicht. Wenn der Primär-Broker in der primären Region ausfällt, können Sie den Replikat-Broker in der sekundären Region zum Primär-Broker hochstufen, indem Sie ein Switchover oder Failover einleiten. Informationen zu den ersten Schritten finden Sie im Entwicklerhandbuch CRDR. Regionsübergreifende Datenreplikation für Amazon MQ für ActiveMQ
18. Mai 2023	Amazon MQ ist jetzt in den folgenden Regionen verfügbar: • Asien-Pazifik (Melbourne) • Asien-Pazifik (Hyderabad) • Europa (Spain) • Europa (Zürich) Informationen über die verfügbaren Regionen finden Sie unter AWS -Regi onen und -Endpunkte im Allgemeinen AWS -Referenzleitfaden.

Datum	Aktualisierung der Dokumentation
14. April 2023	Amazon MQ für RabbitMQ unterstützt jetzt RabbitMQ der Version 3.9.27. Weitere Informationen zu den Korrekturen und Features in diesem Release finden Sie hier: • RabbitMQ 3.9.27 Versionshinweise zum RabbitMQ-Server-Repository GitHub • RabbitMQ-Änderungsprotokoll Weitere Informationen zu den unterstützten Amazon MQ-für-RabbitMQ-Versionen und Broker-Upgrades finden Sie unter Verwalten von Amazon-MQ-für-RabbitMQ-Engine-Versionen.
14. April 2023	Amazon MQ für RabbitMQ unterstützt jetzt RabbitMQ Version 3.10.20. Weitere Informationen zu den Korrekturen und Features in diesem Release finden Sie hier: • RabbitMQ 3.10.20 Versionshinweise zum RabbitMQ-Server-Repository GitHub • RabbitMQ-Änderungsprotokoll Weitere Informationen zu den unterstützten Amazon MQ-für-RabbitMQ-Versionen und Broker-Upgrades finden Sie unter Verwalten von Amazon-MQ-für-RabbitMQ-Engine-Versionen.

Datum	Aktualisierung der Dokumentation
31. März 2023	Amazon MQ für RabbitMQ hat Version 3.10.17 der RabbitMQ-Engine deaktiviert Das Team von Amazon MQ für RabbitMQ und die Open-Source-Maintainer von RabbitMQ haben ein Problem mit der RabbitMQ-Managementkonsole in Version 3.10.17 festgestellt. Amazon MQ hat diese Version zurückgezogen. Um die Auswirkungen dieses Problems zu mildern, erstellen Sie neue Broker mit Version 3.10.20, während wir daran arbeiten, eine neue Patch-Version von RabbitMQ zu unterstützen. Wir empfehlen, die Option Automatisches Unterversion-Upgrade zu aktivieren, um die neuesten Fehlerbehebungen, Sicherheitsupdates und Leistungsverbesserungen automatisch zu erhalten. Weitere Informationen zu verfügbaren Versionen von Amazon MQ für RabbitMQ finden Sie unter Engine-Versionen von Amazon MQ für RabbitMQ.
1. März 2023	Amazon MQ für RabbitMQ unterstützt jetzt RabbitMQ Version 3.10.17. Weitere Informationen zu den Korrekturen und Features in diesem Release finden Sie hier: • RabbitMQ 3.10.17 Versionshinweise zum RabbitMQ-Server-Repository GitHub • RabbitMQ-Änderungsprotokoll Weitere Informationen zu den unterstützten Amazon MQ-für-RabbitMQ-Versionen und Broker-Upgrades finden Sie unter Verwalten von Amazon-MQ-für-RabbitMQ-Engine-Versionen.

Datum	Aktualisierung der Dokumentation
21. Februar 2023	Amazon MQ for RabbitMQ ist jetzt in AWS Key Management Service (KMS) integriert, um serverseitige Verschlüsselung anzubieten. Sie können jetzt Ihren eigenen kundenverwalteten Schlüssel auswählen oder einen CMK verwalteten Schlüssel in Ihrem Konto AWS verwenden. KMS AWS KMS Weitere Informationen finden Sie unter Verschlüsselung im Ruhezustand. Amazon MQ unterstützt die Verwendung von AWS KMS Schlüsseln auf folgende Weise. • Amazon MQ-eigener KMS Schlüssel (Standard) — Der Schlüssel gehört Amazon MQ und wird von Amazon MQ verwaltet und befindet sich nicht in Ihrem Konto. • AWS verwalteter KMS Schlüssel — Der AWS verwaltete KMS Schlüssel (`aws/mq`) ist ein KMS Schlüssel in Ihrem Konto, der in Ihrem Namen von Amazon MQ erstellt, verwaltet und verwendet wird. • Bestehenden, vom Kunden verwalteten KMS Schlüssel auswählen — Vom Kunden verwaltete KMS Schlüssel werden von Ihnen in AWS Key Management Service (KMS) erstellt und verwaltet.
13. Januar 2023	Amazon MQ für RabbitMQ unterstützt jetzt RabbitMQ Version 3.8.34. Weitere Informationen zu den Korrekturen und Features in diesem Release finden Sie hier: • RabbitMQ 3.8.34 Versionshinweise zum RabbitMQ-Server-Repository GitHub • RabbitMQ-Änderungsprotokoll Weitere Informationen zu den unterstützten Amazon MQ-für-RabbitMQ-Versionen und Broker-Upgrades finden Sie unter Verwalten von Amazon-MQ-für-RabbitMQ-Engine-Versionen.

Datum	Aktualisierung der Dokumentation
15. Dezember 202	Amazon MQ für RabbitMQ unterstützt jetzt RabbitMQ der Version 3.9.24. Weitere Informationen zu den Korrekturen und Features in diesem Release finden Sie hier: • RabbitMQ 3.9.24 Versionshinweise zum RabbitMQ-Server-Repository GitHub • RabbitMQ-Änderungsprotokoll Weitere Informationen zu den unterstützten Amazon MQ-für-RabbitMQ-Versionen und Broker-Upgrades finden Sie unter Verwalten von Amazon-MQ-für-RabbitMQ-Engine-Versionen.
13. Dezember 2022	Amazon MQ ist jetzt in der Region Naher Osten (UAE) verfügbar. Informationen über die verfügbaren Regionen finden Sie unter AWS -Regionen und -Endpunkte im Allgemeinen AWS -Referenzleitfaden.
14. November 202	Amazon MQ für RabbitMQ unterstützt jetzt 3.10, eine Hauptversion der Engine. Sie können jetzt Classic Queues Version 2 (CQv2) für Ihre RabbitMQ-Warteschlangen aktivieren. Direkte Updates von 3.8 auf 3.10 werden nicht unterstützt. Weitere Informationen finden Sie hier: • Versionshinweise zu RabbitMQ 3.10.10 • RabbitMQ-Änderungsprotokoll Weitere Informationen zu den unterstützten Amazon MQ-für-RabbitMQ-Versionen und Broker-Upgrades finden Sie unter Verwalten von Amazon-MQ-für-RabbitMQ-Engine-Versionen.

Datum	Aktualisierung der Dokumentation
9. November 2022	Amazon MQ unterstützt jetzt ActiveMQ 5.17.2, eine neue Engine-Unterversion. Weitere Informationen finden Sie hier: • ActiveMQ 5.17.2 – Versionsseite • Verwalten von Amazon MQ für ActiveMQ Engine-Versionen • Aktualisieren einer Amazon MQ-Broker-Engine-Version • Verwenden von XML Spring-Konfigurationsdateien
17. August 2022	Amazon MQ unterstützt jetzt ActiveMQ 5.17.1, eine neue Hauptversion der Engine. Weitere Informationen finden Sie hier: • ActiveMQ 5.17.1 – Versionsseite • Verwalten von Amazon MQ für ActiveMQ Engine-Versionen • Aktualisieren einer Amazon MQ-Broker-Engine-Version • Verwenden von XML Spring-Konfigurationsdateien
14. Juli 2022	Amazon MQ unterstützt jetzt ActiveMQ 5.16.5, eine neue Engine-Unterversion. Weitere Informationen finden Sie hier: • ActiveMQ 5.16.5 – Versionsseite • Verwalten von Amazon MQ für ActiveMQ Engine-Versionen • Verwenden von XML Spring-Konfigurationsdateien • Aktualisieren einer Amazon MQ-Broker-Engine-Version
4. Mai 2022	Amazon MQ fügt inklusive Sprache für das `networkConnector` -Element in der Broker-Konfiguration hinzu. • Erstellen und Konfigurieren eines Amazon MQ-Netzwerks von Brokern

Datum	Aktualisierung der Dokumentation
25. April 2022	Amazon MQ In dieser Version werden der `CRITICAL_ACTION_REQUIRED` Brokerstatus und die `ActionRequired` API Immobilie hinzugefügt. `CRITICAL_ACTION_REQUIRED` informiert Sie, wenn Ihr Broker herunterg efahren ist. `ActionRequired` stellt Ihnen einen Code zur Verfügung, anhand dessen Sie im Entwicklerhandbuch Anweisungen zur Behebung des Problems finden können. • Fehlerbehebung • `ActionRequired` Dokumentation in der Amazon MQ API MQ-Referenz.
20. April 2022	Amazon MQ unterstützt jetzt ActiveMQ 5.16.4, eine neue Engine-Unterversio n. Weitere Informationen finden Sie hier: • ActiveMQ 5.16.4 – Versionsseite • Verwalten von Amazon MQ für ActiveMQ Engine-Versionen • Verwenden von XML Spring-Konfigurationsdateien • Aktualisieren einer Amazon MQ-Broker-Engine-Version
1. März 2022	Amazon MQ ist jetzt in der Region Asien-Pazifik (Jakarta) verfügbar. Informati onen über die verfügbaren Regionen finden Sie unter AWS -Regionen und -Endpunkte im Allgemeinen AWS -Referenzleitfaden.
25. Februar 2022	Amazon MQ für RabbitMQ unterstützt jetzt RabbitMQ Version 3.8.27. Weitere Informationen zu den Korrekturen und Features in diesem Release finden Sie hier: • RabbitMQ 3.8.27 Versionshinweise zum RabbitMQ-Server-Repository GitHub • RabbitMQ-Änderungsprotokoll Weitere Informationen zu den unterstützten Amazon MQ-für-RabbitMQ-Ve rsionen und Broker-Upgrades finden Sie unter Verwalten von Amazon-MQ-für-RabbitMQ-Engine-Versionen.

Datum	Aktualisierung der Dokumentation
16. Februar 2022	Amazon MQ ist jetzt in der Region Afrika (Kapstadt) verfügbar. Informationen über die verfügbaren Regionen finden Sie unter AWS -Regionen und -Endpunkte im Allgemeinen AWS -Referenzleitfaden.
14. Februar 2022	Amazon MQ für RabbitMQ unterstützt jetzt RabbitMQ der Version 3.9.13. Automatische Unterversion-Upgrades können nicht für ein Upgrade von Rabbit 3.8 auf 3.9 verwendet werden. Aktualisieren Sie dazu Ihren Broker manuell. Weitere Informationen zu den neuen Funktionen, die in RabbitMQ 3.9 eingeführt wurden, finden Sie auf der Seite mit den Versionshinweisen für Version 3.9.0 auf der Website. GitHub **Note** Derzeit unterstützt Amazon MQ keine Streams oder die Verwendung der strukturierten Anmeldung, die in JSON RabbitMQ 3.9 eingeführt wurde. Weitere Informationen zu den Korrekturen und Features in diesem Release finden Sie hier: • RabbitMQ 3.9.13 Versionshinweise zum RabbitMQ-Server-Repository GitHub • RabbitMQ-Änderungsprotokoll Weitere Informationen zu den unterstützten Amazon MQ-für-RabbitMQ-Versionen und Broker-Upgrades finden Sie unter Verwalten von Amazon-MQ-für-RabbitMQ-Engine-Versionen.

Datum	Aktualisierung der Dokumentation
07. Februar 2022	Amazon MQ für RabbitMQ führt neue Broker-Metriken ein, mit denen Sie die durchschnittliche Ressourcenauslastung über alle drei Knoten in einer Cluster-Bereitstellung überwachen können. Weitere Informationen finden Sie hier: • the section called “Metriken für RabbitMQ”
18. Januar 2022	Amazon MQ für RabbitMQ unterstützt jetzt RabbitMQ Version 3.8.26. Weitere Informationen zu den Korrekturen und Features in diesem Release finden Sie hier: • RabbitMQ 3.8.26 Versionshinweise zum RabbitMQ-Server-Repository GitHub • RabbitMQ-Änderungsprotokoll Weitere Informationen zu den unterstützten Amazon MQ-für-RabbitMQ-Versionen und Broker-Upgrades finden Sie unter Verwalten von Amazon-MQ-für-RabbitMQ-Engine-Versionen.
13. Januar 2022	Amazon MQ führt den `RABBITMQ_MEMORY_ALARM` -Statuscode ein, um Sie darüber zu informieren, wann Ihr Broker einen Alarm mit hohem Speicher ausgelöst hat und sich in einem ungesunden Zustand befindet. Amazon MQ bietet detaillierte Informationen und Empfehlungen, mit denen Sie hohe Speicheralarme diagnostizieren, auflösen und verhindern können. Weitere Informationen finden Sie unter den folgenden Topics. • the section called “ `RABBITMQ_MEMORY_ALARM` ”

Datum	Aktualisierung der Dokumentation
6. Januar 2022	Wenn Sie CloudWatch Logs for Amazon MQ für ActiveMQ-Broker konfiguri eren, unterstützt Amazon MQ die Verwendung der Kontextschlüssel `aws:SourceArn` und der `aws:SourceAccount` globalen Bedingung sschlüssel in IAM ressourcenbasierten Richtlinien, um das Problem des verwirrten Stellvertreters zu vermeiden. Weitere Informationen finden Sie unter den folgenden Topics. • the section called “Serviceübergreifende Confused-Deputy-Prävention”
20. Dezember 2021	Amazon MQ for ActiveMQ führt eine Reihe neuer Metriken ein, mit denen Sie die maximale Anzahl von Verbindungen überwachen können, die Sie mithilfe verschiedener unterstützter Transportprotokolle mit Ihrem Broker herstellen können, sowie eine zusätzliche neue Metrik, mit der Sie die Anzahl der mit Ihrem Broker verbundenen Knoten in einem Netzwerk von Brokern überwache n können. Weitere Informationen finden Sie unter den folgenden Topics. • the section called “Metriken für ActiveMQ”
16. November 202	Amazon MQ für RabbitMQ unterstützt jetzt RabbitMQ Version 3.8.23. Weitere Informationen zu den Korrekturen und Features in diesem Release finden Sie hier: • RabbitMQ 3.8.23 Versionshinweise zum RabbitMQ-Server-Repository GitHub • RabbitMQ-Änderungsprotokoll Weitere Informationen zu den unterstützten Amazon MQ-für-RabbitMQ-Ve rsionen und Broker-Upgrades finden Sie unter Verwalten von Amazon-MQ-für-RabbitMQ-Engine-Versionen.

Datum	Aktualisierung der Dokumentation
12. Oktober 2021	Amazon MQ unterstützt jetzt ActiveMQ 5.16.3, eine neue Engine-Unterversio n. Weitere Informationen finden Sie hier: • ActiveMQ 5.16.3 – Versionsseite • Verwalten von Amazon MQ für ActiveMQ Engine-Versionen • Aktualisieren einer Amazon MQ-Broker-Engine-Version • Verwenden von XML Spring-Konfigurationsdateien
8. September 2021	Amazon MQ für RabbitMQ unterstützt jetzt RabbitMQ Version 3.8.22. Diese Version enthält einen Fix für ein Problem, bei dem Warteschlangen pro Nachricht TTL (Time to Live) verwendet werden. Dieses Problem wurde in der zuvor unterstützten Version RabbitMQ 3.8.17 identifiziert. Wir empfehlen, Ihre vorhandenen Broker auf Version 3.8.22 zu aktualisieren. Weitere Informationen zu den Updates und Features in diesem Release finden Sie hier: • RabbitMQ 3.8.22 Versionshinweise zum RabbitMQ-Server-Repository GitHub • RabbitMQ-Änderungsprotokoll Weitere Informationen zu den unterstützten Amazon MQ für RabbitMQ-Versionen und Broker-Upgrades finden Sie unter Verwalten von Amazon-MQ-für-RabbitMQ-Engine-Versionen
25. August 2021	Amazon MQ for RabbitMQ hat die RabbitMQ-Engine-Version 3.8.17 vorübergehend deaktiviert, da ein Problem mit Warteschlangen festgestellt wurde, die per-message () verwenden. time-to-live TTL Wir empfehlen die Verwendung der Version 3.8.11.

Datum	Aktualisierung der Dokumentation
29. Juli 2021	Amazon MQ für RabbitMQ unterstützt jetzt RabbitMQ Version 3.8.17. Weitere Informationen zu den in diesem Update enthaltenen Updates und Features finden Sie hier: • RabbitMQ 3.8.17 GitHub Versionshinweise zum RabbitMQ-Server-Repository • RabbitMQ-Änderungsprotokoll • Verwalten von Amazon-MQ-für-RabbitMQ-Engine-Versionen
16. Juli 2021	Sie können jetzt das Wartungsfenster eines Amazon MQ-Brokers mithilfe von AWS Management Console AWS CLI, oder Amazon API MQ anpassen. Weitere Informationen zu Broker-Wartungsfenstern finden Sie hier. • Planung des Wartungsfensters für einen Amazon MQ-Broker
6. Juli 2021	Amazon MQ für RabbitMQ führt die Unterstützung für den Exchange-Typ „Cosistent Hash“ ein. Konsistenter Hash tauscht Routennachrichten an Warteschlangen aus, basierend auf einem Hashwert, der aus dem Routing-Schlüssel einer Nachricht berechnet wird. Weitere Informationen finden Sie hier: • Consistent Hash Exchange Plugin • Konsistenter Hash-Austauschtyp von RabbitMQ im RabbitMQ-Repository GitHub
7. Juni 2021	Amazon MQ unterstützt jetzt ActiveMQ 5.16.2, eine neue Hauptversion der Engine. Weitere Informationen finden Sie hier: • ActiveMQ 5.16.2 – Versionsseite • Verwalten von Amazon MQ für ActiveMQ Engine-Versionen • Aktualisieren einer Amazon MQ-Broker-Engine-Version • Verwenden von XML Spring-Konfigurationsdateien

Datum	Aktualisierung der Dokumentation
26. Mai 2021	Amazon MQ für RabbitMQ ist jetzt in den Regionen China (Peking) und China (Ningxia) verfügbar. Weitere Informationen zu verfügbaren Regionen finden Sie unter AWS Regionen und Endpunkte.
18. Mai 2021	Amazon MQ für RabbitMQ implementiert Broker-Standardwerte. Wenn Sie zum ersten Mal einen Broker erstellen, erstellt Amazon MQ eine Reihe von Broker-Richtlinien und Vhost-Limits basierend auf dem von Ihnen gewählten Instance-Typ und Bereitstellungsmodus, um die Leistung des Brokers zu optimieren. Weitere Informationen finden Sie hier: • Standardwerte für Amazon MQ für RabbitMQ Broker
5. Mai 2021	Amazon MQ unterstützt jetzt ActiveMQ 5.15.15. Weitere Informationen finden Sie hier: • ActiveMQ 5.15.15 – Versionsseite • Verwalten von Amazon MQ für ActiveMQ Engine-Versionen • Verwenden von XML Spring-Konfigurationsdateien
5. Mai 2021	Amazon MQ hat damit begonnen, Änderungen an AWS verwalteten Richtlini en nachzuverfolgen. Weitere Informationen finden Sie hier: • the section called “AWS verwaltete Richtlinien”
14. April 2021	Amazon MQ ist jetzt in den Regionen China (Beijing) und China (Ningxia) verfügbar. Weitere Informationen zu verfügbaren Regionen finden Sie unter AWS Regionen und Endpunkte.
7. April 2021	Amazon MQ unterstützt jetzt RabbitMQ 3.8.11. Weitere Informationen zu den in diesem Update enthaltenen Updates und Features finden Sie hier: • RabbitMQ 3.8.11 Versionshinweise zum RabbitMQ-Server-Repository GitHub • RabbitMQ-Änderungsprotokoll • Verwalten von Amazon-MQ-für-RabbitMQ-Engine-Versionen

Datum	Aktualisierung der Dokumentation
01. April 2021	Amazon MQ ist jetzt in der Region Asien-Pazifik (Osaka) verfügbar. Informationen zu den verfügbaren Regionen finden Sie unter Amazon MQ Regionen und Endpunkte.
21. Dezember 2020	Amazon MQ unterstützt jetzt ActiveMQ 5.15.14. Weitere Informationen finden Sie hier: • ActiveMQ 5.15.14 – Versionshinweise • Verwalten von Amazon MQ für ActiveMQ Engine-Versionen • Verwenden von XML Spring-Konfigurationsdateien • **Important** Aufgrund eines bekannten Apache ActiveMQ Problems in dieser Version wird die neue Pause-Warteschlange in der ActiveMQ Webkonsole kann nicht mit Amazon MQ für ActiveMQ-Broker verwendet werden. Weitere Informationen zu diesem Problem finden Sie unter -8104. AMQ

Datum	Aktualisierung der Dokumentation
4. November 2020	Amazon MQ unterstützt jetztRabbitMQ, ein beliebter Open-Source-Nachri chtenbroker. Auf diese Weise können Sie Ihre vorhandenen RabbitMQ-Nachrichtenbroker zu migrieren, AWS ohne den Code neu schreiben zu müssen. Amazon MQ für RabbitMQ verwaltet sowohl einzelne als auch gecluster te Nachrichtenbroker und übernimmt Aufgaben wie das Bereitstellen der Infrastruktur, das Einrichten des Brokers und das Aktualisieren der Software. • Amazon MQ unterstützt RabbitMQ 3.8.6. Weitere Informationen zu unterstüt zten Engine-Versionen finden Sie unter the section called “Versions verwaltung.”. • Das AWS kostenlose Kontingent umfasst bis zu 750 Stunden einer Einzel-Instance-`mq.t3.micro` -Broker und bis zu 20 GB Speicher pro Monat für ein Jahr. Weitere Informationen zu den unterstützten Instance-Typen finden Sie unter Broker instance types. • Mit Amazon MQ für RabbitMQ können Sie mit AMQP 0-9-1 und in jeder Sprache, die von den RabbitMQ-Clientbibliotheken unterstützt wird, auf Ihre Broker zugreifen. Weitere Informationen zu unterstützten Protokoll en und Cipher Suites finden Sie unter the section called “Amazon MQ für RabbitMQ-Protokolle”. • Amazon MQ für RabbitMQ ist in allen Regionen verfügbar, in denen Amazon MQ derzeit verfügbar ist. Weitere Informationen zu allen verfügbar en Regionen finden Sie in der AWS -Regionentabelle. Informationen zu den ersten Schritten mit Amazon MQ, dem Erstellen eines Brokers und dem Verbinden einer JVM basierten Anwendung mit Ihrem RabbitMQ-Broker finden Sie unter. Erste Schritte: Einen RabbitMQ-Broker erstellen und eine Verbindung zu ihm herstellen

Datum	Aktualisierung der Dokumentation
22. Oktober 2020	Amazon MQ unterstützt ActiveMQ 5.15.13. Weitere Informationen finden Sie hier: • ActiveMQ 5.15.13 – Versionshinweise • Verwalten von Amazon MQ für ActiveMQ Engine-Versionen • Verwenden von XML Spring-Konfigurationsdateien
30. September 202	Amazon MQ ist jetzt in der Region Europa (Mailand) verfügbar. Informationen zu den verfügbaren Regionen finden Sie unter Amazon MQ Regionen und Endpunkte.
27. Juli 2020	Sie können Amazon MQ MQ-Benutzer mit den in Ihrem Active Directory oder einem anderen LDAP Server gespeicherten Anmeldeinformationen authentif izieren. Sie können auch Amazon MQ Benutzer hinzufügen, löschen und ändern und Themen und Warteschlangen Berechtigungen zuweisen. Weitere Informationen finden Sie unter Integrieren von LDAP mit ActiveMQ.
17. Juli 2020	Amazon MQ unterstützt jetzt die `mq.t3.micro` -Instance-Typ. Weitere Informationen finden Sie unter Broker instance types.
30. Juni 2020	Amazon MQ unterstützt ActiveMQ 5.15.12. Weitere Informationen finden Sie hier: • ActiveMQ 5.15.12 – Versionshinweise • Verwalten von Amazon MQ für ActiveMQ Engine-Versionen • Verwenden von XML Spring-Konfigurationsdateien

Datum	Aktualisierung der Dokumentation
30. April 2020	Amazon MQ unterstützt ein neues untergeordnetes Sammlungselement, `systemUsage` , für das Element `broker`. Weitere Informationen finden Sie unter systemUsage. Amazon MQ unterstützt auch drei neue Attribute für das untergeordnete `kahaDB`-Element. • `journalDiskSyncInterval` – Intervall (ms), wann eine Datenträgersynchronisierung durchgeführt werden soll, wenn `journalDiskSyncStrategy=periodic` . • `journalDiskSyncStrategy` – konfiguriert die Richtlinie für die Datenträgersynchronisierung. • `preallocationStrategy` – konfiguriert, wie der Broker versucht, die Journaldateien vorab zuzuweisen, wenn eine neue Journaldatei benötigt wird. Weitere Informationen finden Sie unter Attribute.
3. März 2020	Amazon MQ unterstützt zwei neue Metriken CloudWatch • `TempPercentUsage` – der Anteil des verfügbaren temporären Speichers, der von nicht persistenten Nachrichten verwendet wird. • `JobSchedulerStorePercentUsage` – der Anteil des Festplattenspeichers, der vom Speicher des Aufgaben-Schedulers belegt wird. Weitere Informationen finden Sie unter Monitoring and logging Amazon MQ brokers.
4. Februar 2020	Amazon MQ ist in den Regionen Asien-Pazifik (Hongkong) und Naher Osten (Bahrain) verfügbar Weitere Informationen zu verfügbaren Regionen finden Sie unter AWS Regionen und Endpunkte.

Datum	Aktualisierung der Dokumentation
22. Januar 2020	Amazon MQ unterstützt ActiveMQ 5.15.10. Weitere Informationen finden Sie hier: • ActiveMQ 5.15.10 – Versionshinweise • Verwalten von Amazon MQ für ActiveMQ Engine-Versionen • Verwenden von XML Spring-Konfigurationsdateien
19. Dezember 201	Amazon MQ in den Regionen EU (Stockholm) und Südamerika (São Paulo) verfügbar. Weitere Informationen zu verfügbaren Regionen finden Sie unter AWS Regionen und Endpunkte.

Datum	Aktualisierung der Dokumentation
16. Dezember 201	Amazon MQ unterstützt die Erstellung von durchsatzoptimierten Brokern mithilfe von Amazon Elastic Block Store (EBS) — anstelle des standardm äßigen Amazon Elastic File System (AmazonEFS) — als Broker-Speicher. Verwenden Sie Amazon, um die Vorteile der hohen Haltbarkeit und Replikati on über mehrere Availability Zones hinweg zu nutzenEFS. Verwenden Sie Amazon, um von niedriger Latenz und hohem Durchsatz zu profitierenEBS. **Important** • Sie können Amazon EBS nur mit der `mq.m5` Broker-Instance-Ty pfamilie verwenden. • Obwohl Sie den Broker-Instance-Typ ändern können, ist es nicht möglich, den Speichertyp des Brokers zu ändern, nachdem Sie den Broker erstellt haben. • Amazon EBS repliziert Daten innerhalb einer einzigen Availabil ity Zone und unterstützt den ActiveMQ-Aktiv-/Standby-Bereitstellu ngsmodus nicht. Weitere Informationen finden Sie hier: • Storage • Auswählen des richtigen Broker-Speichertyps für den besten Durchsatz • Die `storageType` Eigenschaft der `broker-instance-op tions` Ressource in der Amazon MQ MQ-Referenz REST API • Die `BurstBalance` -, `VolumeReadOps` - und `VolumeWriteOps` - Metriken im Monitoring and logging Amazon MQ brokers-Abschnitt.
18. Oktober 2019	Zwei CloudWatch Amazon-Metriken sind verfügbar: `TotalEnqueueCount` und`TotalDequeueCount` . Weitere Informationen finden Sie unter Monitorin g and logging Amazon MQ brokers

Datum	Aktualisierung der Dokumentation
11. Oktober 2019	Amazon MQ unterstützt jetzt Endgeräte, die dem Federal Information Processing Standard 140-2 (FIPS) entsprechen, in Handelsregionen der USA. Weitere Informationen finden Sie unter den folgenden Topics: • Bundesstandard für Informationsverarbeitung () 140-2 FIPS • Amazon-MQ-Regionen und -Endpunkte
30. September 2019	Amazon MQ bietet jetzt die Möglichkeit, Ihre Broker durch Ändern des Host-Instance-Typs zu skalieren. Weitere Informationen finden Sie in der `hostInstanceType` -Eigenschaft von `UpdateBrokerInput` und in der `pendingHostInstanceType` -Eigenschaft von `DescribeBrokerOutput`.
30. August 2019	Sie können jetzt die einem Broker zugeordneten Sicherheitsgruppen sowohl in der Konsole als auch mit `UpdateBrokerInput` aktualisieren.
22. Juli 2019	Amazon MQ ist in AWS Key Management Service (KMS) integriert, um serverseitige Verschlüsselung anzubieten. Sie können jetzt Ihren eigenenCMK, vom Kunden verwalteten Schlüssel auswählen oder einen AWS verwalteten KMS Schlüssel in Ihrem AWS KMS Konto verwenden. Weitere Informationen finden Sie unter Verschlüsselung im Ruhezustand. Amazon MQ unterstützt die Verwendung von AWS KMS Schlüsseln auf folgende Weise. • AWS eigener KMS Schlüssel — Der Schlüssel gehört Amazon MQ und befindet sich nicht in Ihrem Konto. • AWS verwalteter KMS Schlüssel — Der AWS verwaltete KMS Schlüssel (aws/mq) ist ein KMS Schlüssel in Ihrem Konto, der in Ihrem Namen von Amazon MQ erstellt, verwaltet und verwendet wird. • Wählen Sie bestehende Kunden aus, verwaltet CMK — Kundenverwaltete CMKs werden von Ihnen in () erstellt und verwaltet. AWS Key Management Service KMS

Datum	Aktualisierung der Dokumentation
19. Juni 2019	Amazon MQ ist in den Regionen Europa (Paris) und Asien-Pazifik (Mumbai) verfügbar. Weitere Informationen zu verfügbaren Regionen finden Sie unter AWS Regionen und Endpunkte.
12. Juni 2019	Amazon MQ ist in der Region Kanada (Zentral) verfügbar. Weitere Informationen zu verfügbaren Regionen finden Sie unter AWS Regionen und Endpunkte.
3. Juni 2019	Zwei neue CloudWatch Amazon-Metriken sind verfügbar: `EstablishedConnectionsCount` und `InactiveDurableSubscribers`. Weitere Informationen finden Sie hier: • Monitoring and logging Amazon MQ brokers • Monitoring and logging Amazon MQ brokers
10. Mai 2019	Der Datenspeicher für neue `mq.t2.micro` -Instance-Typen wurde auf 20 GB beschränkt. Weitere Informationen finden Sie hier: • the section called “Datenspeicherung” • Broker instance types
29. April 2019	Sie können nun Tag-basierte Richtlinien und Berechtigungen auf Ressourcenebene verwenden. Weitere Informationen finden Sie hier: • So funktioniert Amazon MQ mit IAM • Unterstützte Berechtigungen auf Ressourcenebene für Amazon MQ-API-Aktionen
16. April 2019	Sie können jetzt Informationen über die Broker Engine und die Broker-Instance-Optionen mithilfe von abrufen RESTAPI. Weitere Informationen finden Sie hier: • Broker-Instance-Optionen • Broker-Engine-Typen

Datum	Aktualisierung der Dokumentation
8. April 2019	Amazon MQ unterstützt ActiveMQ 5.15.9. Weitere Informationen finden Sie hier: • ActiveMQ 5.15.9 – Versionshinweise • Verwalten von Amazon MQ für ActiveMQ Engine-Versionen • Verwenden von XML Spring-Konfigurationsdateien
4. März 2019	Verbesserte Dokumentation für die Konfiguration des dynamischen Failover und die Anpassung von Clients für ein Netzwerk von Brokern. Aktiviere n Sie das dynamische Failover durch die Konfiguration von `transport Connectors` zusammen mit den `networkConnectors` -Konfigur ationsoptionen. Weitere Informationen finden Sie hier: • Dynamisches Failover mit Transport Connectors • Amazon MQ Brokernetzwerk • Amazon MQ Broker Configuration Parameters
27. Februar 2019	Amazon MQ ist in der Region Europa (London), sowie in folgenden Regionen verfügbar: • Asien-Pazifik (Singapur) • US East (Ohio) • USA Ost (Nord-Virginia) • USA West (Nordkalifornien) • USA West (Oregon) • Asien-Pazifik (Tokio) • Asien-Pazifik (Seoul) • Asien-Pazifik (Sydney) • Europe (Frankfurt) • Europa (Irland)
24. Januar 2019	Die Standardkonfiguration enthält jetzt eine Richtlinie zum Löschen inaktiver Ziele.

Datum	Aktualisierung der Dokumentation
17. Januar 2019	Amazon MQ `mq.t2.micro` -Instance-Typen unterstützen jetzt nur 100 Verbindungen pro Wire-Level-Protokoll. Weitere Informationen finden Sie unter Quotas in Amazon MQ.
19. Dezember 2018	Sie können eine Reihe von Amazon MQ-Brokern in einem Netzwerk von Brokern konfigurieren. Weitere Informationen finden Sie in den folgenden Abschnitten: • Amazon MQ Brokernetzwerk • Creating and Configuring a Network of Brokers • Korrekte Konfiguration Ihres Netzwerk von Brokern • `networkConnector` • `networkConnectionStartAsynchron`

11. Dezember 2018	Amazon MQ unterstützt ActiveMQ 5.15.8, 5.15.6 und 5.15.0.
	• Fehlerbehebungen und Verbesserungen in ActiveMQ:
	  • ActiveMQ 5.15.8 – Versionshinweise
	  • ActiveMQ 5.15.7 – Versionshinweise

5. Dezember 2018	AWS unterstützt die Kennzeichnung von Ressourcen, damit Sie Ihre Kostenzuweisung verfolgen können. Sie können Ressourcen beim Erstellen oder durch Anzeigen der Details dieser Ressource markieren. Weitere Informationen finden Sie unter Markieren von Ressourcen.
19. November 2018	AWS hat sein SOC Compliance-Programm um Amazon MQ als SOCkonformen Service erweitert.
15. Oktober 2018	• Die maximale Anzahl an Gruppen pro Benutzer ist 20. Weitere Informationen finden Sie unter Benutzer. • Die maximale Anzahl an Verbindungen pro Broker pro Wire-Level-Protokoll ist 1 000. Weitere Informationen finden Sie unter Broker.
2. Oktober 2018	AWS hat sein HIPAA Compliance-Programm um Amazon MQ als HIPAAberechtigten Service erweitert.

Datum	Aktualisierung der Dokumentation

27. September 2018	Amazon MQ unterstützt ActiveMQ 5.15.6, zusätzlich zu 5.15.0. Weitere Informationen finden Sie hier:
	• Erste Schritte: Einen ActiveMQ-Broker erstellen und eine Verbindung zu ihm herstellen
	• Fehlerbehebungen und Verbesserungen in der ActiveMQ-Dokumentation:
	  • ActiveMQ 5.15.6 – Versionshinweise
	  • ActiveMQ 5.15.5 – Versionshinweise
	  • ActiveMQ 5.15.4 – Versionshinweise
	  • ActiveMQ 5.15.3 – Versionshinweise
	  • ActiveMQ 5.15.2 – Versionshinweise
	  • ActiveMQ 5.15.1 – Versionshinweise
	• ActiveMQ-Client 5.15.6

31. August 2018	• Die folgenden Metriken sind verfügbar:
	  • `CurrentConnectionsCount`
	  • `TotalConsumerCount`
	  • `TotalProducerCount`
	  Weitere Informationen finden Sie im Abschnitt Monitoring and logging Amazon MQ brokers.
	• Die IP-Adresse des Brokers wird auf der Seite Details angezeigt.
	  Note
	  Für Broker mit deaktivierter öffentlicher Zugänglichkeit wird die interne IP-Adresse angezeigt.

Datum	Aktualisierung der Dokumentation
30. August 2018	Amazon MQ ist in der Region Asien-Pazifik (Singapur), sowie in folgenden Regionen verfügbar: • US East (Ohio) • USA Ost (Nord-Virginia) • USA West (Nordkalifornien) • USA West (Oregon) • Asien-Pazifik (Tokio) • Asien-Pazifik (Seoul) • Asien-Pazifik (Sydney) • Europe (Frankfurt) • Europa (Irland)
30. Juli 2018	Sie können Amazon MQ so konfigurieren, dass allgemeine Protokolle und Auditprotokolle in Amazon Logs veröffentlicht CloudWatch werden. Weitere Informationen finden Sie unter Monitoring and logging Amazon MQ brokers.
25. Juli 2018	Amazon MQ ist in den Regionen Asien-Pazifik (Tokio) und Asien-Pazifik (Seoul), sowie in folgenden Regionen verfügbar: • US East (Ohio) • USA Ost (Nord-Virginia) • USA West (Nordkalifornien) • USA West (Oregon) • Asien-Pazifik (Sydney) • Europe (Frankfurt) • Europa (Irland)
19. Juli 2018	Sie können es verwenden AWS CloudTrail , um Amazon MQ API MQ-Anrufe zu protokollieren. Weitere Informationen finden Sie unter Logging Amazon MQ API calls using CloudTrail.

Datum	Aktualisierung der Dokumentation
29. Juni 2018	Zusätzlich zu `mq.t2.micro` und `mq.m4.large` stehen die folgenden Broker-Instance-Typen für reguläre Entwicklungs-, Test- und Produktions-Workloads zur Verfügung, die einen hohen Durchsatz erfordern: • `mq.m5.large` • `mq.m5.xlarge` • `mq.m5.2xlarge` • `mq.m5.4xlarge` Weitere Informationen finden Sie unter Broker instance types.
27. Juni 2018	Amazon MQ ist in der Region USA West (Nordkalifornien), sowie in folgenden Regionen verfügbar: • US East (Ohio) • USA Ost (Nord-Virginia) • USA West (Oregon) • Asien-Pazifik (Sydney) • Europe (Frankfurt) • Europa (Irland)

Datum	Aktualisierung der Dokumentation

14. Juni 2018	• Sie können die `AWS::Amazon MQ::Broker` AWS CloudFormation Ressource verwenden, um die folgenden Aktionen auszuführen:
	  • Erstellen eines Brokers.
	  • Hinzufügen von Konfigurationsänderungen sowie Bearbeiten von Benutzern für den angegebenen Broker.
	  • Rückgabe von Informationen über den angegebenen Broker.
	  • Löschen des angegebenen Brokers.
	
	  Note
	  Wenn Sie eine Eigenschaft des Eigenschaftstyps Amazon MQ Broker ConfigurationId oder Amazon MQ Broker-Benutzer ändern, wird der Broker sofort neu gestartet.
	• Sie können die `AWS::Amazon MQ::Configuration` AWS CloudFormation Ressource verwenden, um die folgenden Aktionen durchzuführen:
	  • Erstellen einer Konfiguration.
	  • Aktualisieren der angegebenen Konfiguration.
	  • Rückgabe von Informationen über die angegebene Konfiguration.
	
	  Note
	  Sie können AWS CloudFormation damit eine Amazon MQ MQ-Konfiguration ändern, aber nicht löschen.

7. Juni 2018	Die Amazon MQ-Konsole unterstützt Deutsch, Brasilianisches Portugiesisch, Spanisch, Italienisch und Traditionelles Chinesisch.
17. Mai 2018	Die Anzahl der Benutzer pro Broker ist auf 250 begrenzt. Weitere Informationen finden Sie unter Benutzer.
13. März 2018	Das Erstellen eines Brokers dauert ca. 15 Minuten. Weitere Informationen finden Sie unter Abschließen der Broker-Erstellung.

Datum	Aktualisierung der Dokumentation
1. März 2018	• Sie können die gleichzeitige Speicherung und Bereitstellung für Apache KahaDB mit dem Attribut `concurrentStoreAndDispatchQueues` konfigurieren. • Die `CpuCreditBalance` CloudWatch Metrik > ist für den Broker-In stance-Typ verfügbar. `mq.t2.micro`
10. Januar 2018	Die folgenden Änderungen wirken sich auf die Amazon MQ-Konsole aus: • In der Liste der Broker ist die Spalte Erstellung standardmäßig ausgeblen det. Wählen Sie zum Anpassen der Seitengröße und der Spalten ⚙ aus. • Auf dem ***MyBroker***Wählen Sie auf der Seite im Abschnitt Verbindungen den Namen Ihrer Sicherheitsgruppe oder öffnen Sie die EC2 Konsole (statt der VPC Konsole). Die EC2 Konsole ermöglicht eine intuitivere Konfiguration von Regeln für eingehenden und ausgehenden Datenverkehr. Weitere Informationen finden Sie im aktualisi erten Abschnitt Eingehende Verbindungen aktivieren.
9. Januar 2018	• Die Berechtigung für die REST Vorgangs-ID `UpdateBroker` ist wie `mq:UpdateBroker` auf der IAM Konsole korrekt aufgeführt. • Die irrtümliche `mq:DescribeEngine` Berechtigung wurde aus der IAM Konsole entfernt.

Datum	Aktualisierung der Dokumentation

28. November 2017	Dies ist die erste veröffentlichte Version von Amazon MQ und des Amazon MQ Entwicklerhandbuchs.
	• Amazon MQ ist in den folgenden Regionen verfügbar:
	  • US East (Ohio)
	  • USA Ost (Nord-Virginia)
	  • USA West (Oregon)
	  • Asien-Pazifik (Sydney)
	  • Europe (Frankfurt)
	  • Europa (Irland)
	  Die Nutzung des *mq.t2.micro* Instance-Typs ist abhängig von CPUCredits und Baseline-Performance — mit der Möglichkeit, das Baseline-Level zu übertreffen (weitere Informationen finden Sie in der *CpuCreditBalance* Metrik). Wenn Ihre Anwendung Fixed Performance, erwägen Sie, einemq.m5.large -Instance-Typ zu verwenden.
	• Sie können mq.m4.large - und mq.t2.micro -Broker erstellen.
	  Die Nutzung des mq.t2.micro Instance-Typs ist abhängig von den CPUCredits und der Baseline-Performance — mit der Möglichkeit, das Baseline-Level zu übertreffen (weitere Informationen finden Sie in der CpuCreditBalance Metrik). Wenn Ihre Anwendung Fixed Performance, erwägen Sie, einen mq.m5.large -Instance-Typ zu verwenden.
	• Sie können die Broker-Engine ActiveMQ 5.15.0 verwenden.
	• Sie können Broker auch programmgesteuert mit Amazon RESTAPIMQ und verwalten. AWS SDKs
	• Sie können auf Ihre Broker zugreifen, indem Sie jede Programmiersprache verwenden, die ActiveMQ unterstützt, und indem Sie sie TLS explizit für die folgenden Protokolle aktivieren:
	  • AMQP
	  • MQTT
	  • MQTTüber WebSocket
	  • OpenWire

Datum	Aktualisierung der Dokumentation
	• STOMP • STOMPüber WebSocket • Sie können unter Verwendung verschiedener ActiveMQ-Clients eine Verbindung zu ActiveMQ-Brokern einrichten. Wir empfehlen die Verwendung des ActiveMQ-Clients. Weitere Informationen finden Sie unter Connecting a Java application to your broker. • Ihr Broker kann Nachrichten in beliebiger Größe versenden und empfangen.

Die vorliegende Übersetzung wurde maschinell erstellt. Im Falle eines Konflikts oder eines Widerspruchs zwischen dieser übersetzten Fassung und der englischen Fassung (einschließlich infolge von Verzögerungen bei der Übersetzung) ist die englische Fassung maßgeblich.